1. CCS3.3 DSP/BIOS SWI(Sofyware Interrupt Manager) 생성

1. 디렉토리 구성

₩cmd	: Linker 컴맨드 파일
DSP2833x_headers	: Chip관련 헤더 파일 및 헤더용 Linker 컴맨드 파일
₩include	: 사용자 인쿠르드 파일
₩testprj_3	: 사용자 프로젝트 파일 및 실행 파일(.HEX)
₩testsrc_3	: 사용자 소스 파일

2. 디렉토리 설명

다른 디렉토리는 기존 Task 생성 예제 파일을 사용 하시고 testprj_3, testsrc_3를 복사 후 include 에 있는 *.h 파일을 복사해서 사용 하시면 됩니다.

3. Setup CCStudio v3.3을 실행 합니다. 프로젝트 Open 사용법은 Task 생성 예제를 참조 하시기 바랍니다.

4. 소스코드 설명(Main_Bios.c)

#include "DSP2833x_Device.h" #include "F28335_example.h"	<- DSP 초기화 및 설정 관련 <- 사용자 외부 함수,변수,정의 관리
void main(void) {	
InitSysCtrl(); InitPieCtrl(); InitWatchdog(); InitGpio(); InitXintf();	<- CPU 클럭 설정((30*10) / 2 = 150M) <- 인터럽트 관련 초기화 <- watch-dog 설정 및 초기화 <- CPU I/O 설정(IN,OUT,기본기능) _EX_BUS_ON정의에 따라 외부 버스 ON <- 내부 주변 디바이스 클럭 설정 및 외부 버스 타이밍 설정
** DSP/BIOS 관련 설정 ** ifdef EXAMPLE_FLASH memcpy(&secureRamFunc &secureRamFunc InitFlash(); #endif	s_runstart, cs_loadstart,&secureRamFuncs_loadend - &secureRamFuncs_loadstart);
// Peripheral Initialization InitAdc(); InitEPwm();	<- ADC CH0 초기화및 인터럽트 On <- ADC 소스 클럭 공급 PWM 초기화
asm(" EALLOW"); GpioCtrlRegs.GPBMUX1.bit GpioCtrlRegs.GPBDIR.bit.G GpioDataRegs.GPBSET.bit. asm(" EDIS");	<- Enable EALLOW protected register access t.GPIO32 = 0;<- GPIO032 GPIO GPIO32 = 1; <- GPIO32 output .GPIO32 = 1; <- GPIO32 pin is set to 1 // Disable EALLOW protected register access
** DSP/BIOS에서 TINT2,DLOGIN SetDBGIER(IER 0x6000); *(volatile unsigned int *)0xi	IT를 사용 하므로 BIOS사용 인터럽트 허가 ** <- Enable everything in IER, plus TINT2 and DLOGINT 00000C14 = 0x0C00;<- Set TIMER2 FREE=SOFT=1
** 아래 main()를 종료 하면 DSP/	'BIOS가 동작 **
}	
void UserInit(void){	<- 이 함수는 리셋시 DSP/BIOS 초기화 부분에서 한번 수행 후 DSP/BIOS관련 및 사용자 초기화 함수 추가
}	
void task1_proc(void){	<- 이 함수는 스케줄러에 관리 되는 TASK
}	
5. 소스코드 설명(DefaultIsr_BIOS.c	
void ADCINT_ISR(void){	<- ADC H/W(P1_6) 인터럽트를 바로 처리하지 않고 SWI로 변환하여 실행.
PieCtrlRegs.PIEACK.all = P	IEACK_GROUP1; <- Must acknowledge the PIE group
SWI_post(&ADC_swi); }	<- SWI 로 변환

- CCS3.3 DSP/BIOS SWI생성

* SWI란 하드웨어 인터럽트에서 발생한 인터럽트를 바로 처리 하지 않고 SWI 인터럽트를 처리하는 함수 (TASK)를 미리 생성해 그곳에서 인터럽트를 처리 하도록 하는 기능이 있다. HWI 루틴에서는 SWI_post() 함수를 이용 한다, 일반적으로 인터럽트가 발생하면 바로처리 하는데 반해 SWI가 있는 이유는 BIOS에서 다른 TASK를 처리할 시간이 모자라므로 인터럽트도 TASK화 하여 관리 하기 위함이다. 물론 SWI인터럽트를 사용 하지 않을 수도 있다.



1. Main_Bios.c 를 open후 아래 소스코드를 입력 후 저장한다.

void AdcSwi(void) <- SWI 스케줄 함수 명 :: 실제 ADC DATA 부분 처리 { static Uint16 *AdcBufPtr = AdcBuf; <- ADC 저장 값 버퍼 // Manage the ADC registers AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1; <- Reset SEQ1 to CONV00 state AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1; <- Clear ADC SEQ1 interrupt flag // Read the ADC result *AdcBufPtr++ = AdcRegs.ADCRESULT0 >> 4; <- ADC 데이터 저장 // Brute-force the circular buffer if(AdcBufPtr == (AdcBuf + ADC_BUF_LEN)) <- 버퍼 초과 검사 Ring Buf 구현 AdcBufPtr = AdcBuf; 2. DSP/BIOS Config->*.tcf 를 실행 한다.



3. 하드웨어 인터럽트를 SWI로 변환(Scheduling ->HWI Hardware Interrupt Service Routine Manager)



* 하드웨어 인터럽트 등록(HWI)

PIE_INT1_6 등록 정보			
General Dispatcher			
comment: defines function for the P			
function: _ADCINT_ISR			호출 되는 함수 명 앞에 _를 붙임
monitor Nothing -			
addr: 0x00000000			
type signed 💌			
operation STS_add(*addr) 💌			
확인 취소	적용(<u>A</u>)	도움말	

* 소프트웨어 인터럽트 허가(SWI)

PIE_INT1_6 등록 정보			
General Dispatcher	1		_ 체크 하지 않으면 HWI, 체크하면 SWI 허가
✓ Use Dispatcher			
Arg:	0x0000000		
Interrupt Mask IER0	self		
Interrupt Bit Mask IER:	0x0001		
확인	취소 적용(<u>A</u>)	도움말	

4. Scheduling SWI 등록(. Scheduling ->Software Interrupt Manager)



* SWI 관리 명을 입력 한다.(프로그램 에서 호출 주소가 됨.)

Insert Object				
Please enter a	name for the Confi	guration		
	ОК		입력 후	확인
ADC_SWI	ОК		입력 후	

- * Task 관리자 생성 확인
 - System
 Instrumentation
 CLK Clock Manager
 PRD Periodic Function Manager
 HWI Hardware Interrupt Service Routine Manager
 KNL Software Interrupt Manager
 KNL_swi
 PRD_swi
 TSK Task Manager
 IDL Idle Function Manager
 Synchronization
 Input/Output
- * 생성한 LOG tace에 사용자 환경을 설정 한다.(trace 선택후 오른쪽 버튼)



* General 에서 기본 정보를 설정 한다.

ADC_swi 등록 정보	
General	
comment: <add comments="" here=""></add>	
function: _AdcSwi	— 호출 되는 함수 명 앞에 를 붙임
priority 2	호출 되는 함수 우선순위 레벌(1(저)-15(고))
mailbox: 0	
arg0: 0x0000000	
arg1: 0x0000000	
확인 취소 적용(<u>A</u>) 도움말	

* Scheduling SWI에서 생성 된 ADC_swi를 확인후 *.tcf 파일을 종료 합니다.

F28335_example_BIOS_flash.tcf *			
Estimated Data Size: 1278 Est. Min. Stack Size (MAUs): 254	ADC_swi prope	rties	Textual Configuration Script - Line: 1
 System Instrumentation Scheduling CLK - Clock Manager PRD - Periodic Function Manager HWI - Hardware Interrupt Service Routine Manager SWI - Software Interrupt Manager SWI - Software Interrupt Manager KNL_swi PRD_swi TSK - Task Manager JIDL - Idle Function Manager Synchronization Input/Output 	Property comment function priority mailbox arg0 arg1	Value <add comm<br="">_AdcSwi 1 0 0x00000000 0x00000000</add>	<pre>utils.loadPlatform("ti.platforms.ezdsp28: /* The following DSP/BIOS Features are er bios.enableRealTimeAnalysis(prog); bios.enableRtdx(prog); bios.enableRtdx(prog); bios.GBL.CLKIN = 30000; bios.GBL.CLLUSERINITFXN = 1; bios.GBL.USERINITFXN = prog.extern("User] bios.GBL.WODIFYPLCR2 = 0; bios.MEM.NONEMORYHEAPS = 0; bios.MEM.instance("ZONE7"); bios.MEM.instance("ZONE7"); bios.MEM.instance("ZONE7").lose = 0x2000; bios.MEM.instance("ZONE7").lose = 0x2000; bios.MEM.instance("ZONE7").lose = 0x2000; bios.MEM.instance("ZONE7").lose = code' bios.MEM.instance("ZONE7").space = "code' bios.MEM.instance("FLASH").destroy(); tios.MEM.instance("FLASH").destroy(); tios.MEM.instance("DOPERGUNC.")</pre>



- CCS3.3 DSP/BIOS PRD TASK 실행

1. JTAG 및 에뮬레이터를 연결 합니다.

😻 /F28335 XDS510USB	Emulator/cpu_0 - 1	TMS320C28xx - Code
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject	<u>Debug</u> <u>G</u> EL <u>Option</u>	P <u>r</u> ofile <u>T</u> ools D <u>S</u> P/BI
쒑 🚅 🔲 🐰 🖻 💼	<u>B</u> reakpoints	
test, pjt	Assembl <u>y</u> /Source St Step <u>I</u> nto	epping F11
🔊 60° 📋 🔜 🔜	Step <u>O</u> ver Step O <u>u</u> t	F10 Shift+F11
?? ??	<u>Bun</u> Halt Animate Run <u>F</u> ree Low Power Run Run to Cursor Set PC to Cursor R <u>e</u> start <u>G</u> o Main Multiple Operation,	F5 Shift+F5 Alt+F5 CtrI+F5 CtrI+Alt+F5 CtrI+F10 CtrI+Shift+F10 CtrI+Shift+F5 CtrI+Shift+F5
※ ※	Adva <u>n</u> ced Resets Reset CPU Reset Emulator Halt on Reset	► Ctrl+R Ctrl+Shift+R
□□	<u>C</u> onnect Restore <u>D</u> ebug State	Alt+C 🔶

2. 내부램 으로 프로그램을 실행할 경우 아래와 같이 설정 합니다.(Option->Customize)

Customize 🛛 🛛 🗙	
Debug Properties Directories Color Keyboard Program/Project/Cl0 Control Window Dist	
Program Load ✓ Perform verification during Program Load ✓ Load Program After Build ✓ Do Not Set CIO Breakpoint At Load ✓ Do Not Set End Of Program Breakpoint At Load ✓ Disable All Breakpoints When Loading New Programs	체크
Project Image: Open Dependent Projects When Loading Projects Image: Do Not Scan Dependencies When Loading Projects Image: Auto-save Projects Before Build Image: Open Project Window On Startup	
CIO Maximum # of lines: 256 (may require restart)	
확인 취소 적용(<u>A</u>) 도움말	

3. 컴파일 하기(Project->Rebuild All)

휋 /F28335 XDS51	LOUSB Emul	ator/cpu_0	- TMS32)C28xx -	Code C
<u>F</u> ile <u>E</u> dit <u>V</u> iew	Project D	ebug <u>G</u> EL	<u>O</u> ption	P <u>r</u> ofile	<u>T</u> ools
[™]	<u>N</u> ew <u>O</u> pen Us <u>e</u> Exte	rnal Makefi	le		i ##
응산 월 (%) (%) (%)	E <u>x</u> port to <u>A</u> dd File <u>S</u> ave <u>C</u> lose	o Makefile s to Project			
r∰ L <mark>@</mark> f.	So <u>u</u> rce (Control		I	
??) □	Compile <u>B</u> uild	<u>F</u> ile	i I	Ctrl + F 7 F 7	
	<u>R</u> ebuild	All			



에러 확인

4. FLASH에 프로그램 하기

۰ 😻	/F2833	5 XDS	51 OUSE	Emula	tor/c	ou _0 -	TMS32	JC28xx	- Code C	ompose	r Studio	- Not
<u> </u>	<u>E</u> dit	<u>V</u> iew	<u>P</u> roject	<u>D</u> ebug	<u>G</u> EL	<u>O</u> ption	Profile	<u>T</u> ools	D <u>S</u> P/BIOS	<u>W</u> indow	<u>H</u> elp	
- *	🖻 F	ЦX	e Ri	in al				F28x	x On-Chip F	lash Progr	ammer	N ?
								Data	Converter S	upport		· ·

* 아래 CLOCK 설정 메뉴를 사용자에 맞게 설정 합니다.

Clock Configuration		×
OSCCLK	30	OK
DIVSEL:	/2 💌	Cancel
PLLCR Value:	10 -	
SYSCLKOUT	150,0000	

* API Interface 파일을 등록 합니다.

Flash Programmer Settings			
Select DSP Device to Program	Options ↓ Load Symbols ↓ Display Tooltips ↓ Display Diagnostics ↓ Save Clock Settings	OK Cancel Help	
Select version of Flash API Interface			
		Browse	

열기				? 🗙
찾는 위치([):	28335	•	+ 🗈 💣 🎟 -	
FlashAPIInterfac	ce28335V1_01,out			
파일 이름(<u>N</u>):	FlashAPIInterface28335V2_10,out		_	기(<u>0</u>)
파일 형식(<u>T</u>):	COFF Files (*,out)		▼ ⁵	i소

Flash Programmer Settings				
Select DSP Device to Program	Options Load Symbols Display Tooltips Display Diagnostics Save Clock Settings	OK Cancel Help		
Select version of Flash API Interface				





On-Chip Flash Programmer		닫음
On-Chip Flash Programmer Clock Configuration OSCCLK 30 DIVSEL: /2 PLLCR Value: 10 SYSCLKOUT 150,0000 Code Security Password Key 7 FFFF Key 6 FFFF Key 7 FFFF Key 3 FFFF Key 2 FFFF Key 1 FFFF Key 0 FFFF Key 0 FFFF Value: 1 Value: 1 Key 1 FFFF Key 0 FFFF Value: 1 Key 1 FFFF Key 0 FFFF Value: 1 Value: 1 Value: 1 Value: 1 Key 1 FFFF Key 0 FFFF Value: 1 Value: 1 Value: 1 Value: 1 Key 1 FFFF Key 0 FFFF <td>Erase Sector Selection ▼ Sector A: (338000-33FFF) ▼ Sector F: (310000-317FFF) ▼ Sector B: (330000-337FFF) ▼ Sector G: (308000-30FFFF) ▼ Sector C: (328000-32FFF) ▼ Sector H: (300000-307FFF) ▼ Sector D: (320000-327FFF) ▼ Sector I: (3DA000-3D8FFF) ▼ Sector E: (318000-31FFFF) ▼ Sector I: (3DA000-3D8FFF) ▼ Sector E: (318000-31FFFF) ▼ Sector J: (3D8000-3D9FFF) ● Operation Please specify the COFF file to □D:₩lang₩cpuplaza₩dsp28335_bios₩testprj_1₩Debugħ Browse, • Erase Only ● Frequency Test • Program, Verify Register: • Program Only Pin: • Program Only Pin: • Flash Random Wait State: 15 • Calculate Checksums Flash Page Wait State: • TP Wait State: 31 • OTP: • Load RAM Only Flash+0TP: • Execute Operation Help <td>00 57</td></td>	Erase Sector Selection ▼ Sector A: (338000-33FFF) ▼ Sector F: (310000-317FFF) ▼ Sector B: (330000-337FFF) ▼ Sector G: (308000-30FFFF) ▼ Sector C: (328000-32FFF) ▼ Sector H: (300000-307FFF) ▼ Sector D: (320000-327FFF) ▼ Sector I: (3DA000-3D8FFF) ▼ Sector E: (318000-31FFFF) ▼ Sector I: (3DA000-3D8FFF) ▼ Sector E: (318000-31FFFF) ▼ Sector J: (3D8000-3D9FFF) ● Operation Please specify the COFF file to □D:₩lang₩cpuplaza₩dsp28335_bios₩testprj_1₩Debugħ Browse, • Erase Only ● Frequency Test • Program, Verify Register: • Program Only Pin: • Program Only Pin: • Flash Random Wait State: 15 • Calculate Checksums Flash Page Wait State: • TP Wait State: 31 • OTP: • Load RAM Only Flash+0TP: • Execute Operation Help <td>00 57</td>	00 57
Flash Programmer Settings		
•	▼ ↓	



5. 프로그램을 로딩후 Debug 탭에서 Go Main 기능을 실행 합니다.

6. Main_Bios.c의 AdcSWI() 에 break mode를 설정 합니다.

```
// [인수] void
// [참고] DSP/BIOS 500ms 스케줄 타임
// -----
                              _____
void AdcSwi(void)
                                        // Pointer to buffer
static Uint16 *AdcBufPtr = AdcBuf;
// Manage the ADC registers
   AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1; // Reset SEQ1 to CONVOO state
AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1; // Clear ADC SEQ1 interrupt flag
// Read the ADC result
                                        // Read the result
   *AdcBufPtr++ = AdcRegs.ADCRESULTO >> 4;
22 Brute-force the circular buffer
   if( AdcBufPtr == (AdcBuf + ADC_BUF_LEN) )
      AdcBufPtr = AdcBuf;
                                 // Rewind the pointer to beginning
}
```

- F5 : Debuf->Run

- F9 : Debuf->Break Point Toggle
- 7. DSP BIOS Message Log(DSP/BIOS->Message Log) 예제 프로그램에서 LOG_printf()로 출력되는 멧세지를 확인 할수 있다.

😻 /F28335 XDS510USB Emulator/cpu_0 - TMS320C28xx - Code Comj	poser Studio - [Main_BIOS.c]	
♦ Eile Edit View Project Debug GEL Option Profile Tools	D <u>S</u> P/BIOS <u>W</u> indow <u>H</u> elp	
🎽 🚅 📕 👗 🛍 💼 🗠 여 여	<u>R</u> TA Control Panel	⊜ №
F28335_example_BIOS 💌 Debug 💽 😒 🛗 📇 🝝	Statistics View	
🔊 & 🗅 🖩 🛤 🖩 🗖 🗗 🌮	<u>M</u> essage Log	
Image: Control of the state	Kernel/Object View <u>H</u> ost Channel Control <u>C</u> PU Load Graph	***** 3x_Dev _examp
Message Log	×	
4 GPI0 Blick Count = 4		
5 GPI0_Blink Count = 5	_	lo_Blink() 함수에서 표시
6 GPI0_Blink Count = 6		
7 GPIU_Blink Count = 7 8 GPIU_Blink Count = 8	=	
9 GPI0_Blink Count = 9		
10 GPI0_Blink Count = 10		
11 GPIU_Blink Count = 11 12 GPIO_Blink Count = 12		
13 GPIO_Blink Count = 13		
14 GPIO Blink Count = 14		
15 GPI0_Blink Count = 15	-	
IC CDIO DELL'OSTAN IC		

8. A/D 데이터 확인(데이터 버퍼)

- // 변수 <u>선언</u>		
Uint16 AdcB	Editor	▶ 오른쪽 마우스(AdcBuf)
	Add to Watch Window	◀━━━ 선택(왼쪽 마우스)
	Quick Watch	
	Open Document	
	View Location of AdcBuf	
	<u>M</u> ixed Mode	
	Cu <u>t</u>	
	<u>C</u> opy	
	<u>P</u> aste	
	Select A <u>l</u> l	
	<u>S</u> et PC to Cursor	
	<u>R</u> un To Cursor	
	Toggle Software Breakpoint	
	Toggle Hardware Breakpoint	
	B <u>o</u> okmarks	•
	Insert Graph	
	<u>G</u> o to	
	Advanced	•
-	0	

Watch Windo	w		×
Name	Value	Туре	Radix 🔺
🖃 🔗 Ad	0×0000C	unsigned int,	hex
↓ [0]	4095	Uint16	unsigned
I (1)	4095	Uint16	unsigned
[2]	4095	Uint16	unsigned
🔶 [3]	4095	Uint16	unsigned
[4]	4095	Uint16	unsigned 📃
§ [5]	4095	Uint16	unsigned
🔶 [6]	0	Uint16	unsigned
	19	Uint16	unsigned
🔶 [8]	0	Uint16	unsigned
🔶 [9]	1	Uint16	unsigned
I I I I I I I I I I I I I I I I I I I	0	Uint16	unsigned
♦ [0	Uint16	unsigned
♦ [8	Uint16	unsigned
♦ [0	Uint16	unsigned
♦ [3	Uint16	unsigned
♦ [0	Uint16	unsigned
	0	Uint16	unsigned 👻
🙎 Watch Lo	cais 🔗 Wa	atch 1	



🐱 Graph Property Dialog		
Display Type	Single Time 🔺 🔒	_ 그레프 종류
Graph Title	Graphical Display	
Start Address	AdcBuf	- 보고자 하는 버퍼명
Page	Data	
Acquisition Buffer Size	50 🔸	- 버퍼 크기
Index Increment	1	
Display Data Size	50 🗧	- 버퍼 크기
DSP Data Type	16-bit unsigned integer 🔸 🚽 🚽	- 데이터 형
Q-value	0	
Sampling Rate (Hz)	50000 ┥	 A/D Sample Rate
Plot Data From	Left to Right	
Left-shifted Data Display	Yes	
Autoscale	On	
DC Value	0	
Axes Display	On	
Time Display Unit	us 🖕 📃 🔻	- 표시 단위 설정
	<u>O</u> K <u>C</u> ancel <u>H</u> elp	



10. A/D 데이터 리얼 타임 표시 - 항목8,9를 실시간으로 모니터링 하기 위한 방법이다. ** 주의 아래 순서를 지키지 않으면 장비에 심각한 문제가 발생 할수 있다 **

- * START
- 1. 프로그램을 로딩 한다.
- 2. ccs3.3 Debug->Halt 상태나, STOP 상태로 만든다.
- 3. Debug->Real-Time_Mode를 선택하여 체크 한다.
- 4. View->Real Time Options->Global Continuous를 선택하여 체크 후 OK 버튼 클릭.
- 5. 8번 이나, 9번 항목 실행 :: 데이터 보기

* STOP

- 1. Debug->Halt 를 실행 한다.
- 2. Debug->Real-Time_Mode를 해제 한다.