# 1. CCS3.3 DSP/BIOS PRD(periodic fuction manager) 생성

1. 디렉토리 구성

₩cmd	: Linker 컴맨드 파일
DSP2833x_headers	: Chip관련 헤더 파일 및 헤더용 Linker 컴맨드 파일
₩include	: 사용자 인쿠르드 파일
₩testprj_2	: 사용자 프로젝트 파일 및 실행 파일(.HEX)
₩testsrc_2	: 사용자 소스 파일

2. 디렉토리 설명

다른 디렉토리는 기존 Task 생성 예제 파일을 그대로 사용하고 testprj\_2, testsrc\_2 디렉토리만 복사에서 사용 하거나 기존의 testprj\_1,testsrs\_1 의 소스만 아래 진행에 따라 추가 수정 하여 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);
asm(" EALLOW"); GpioCtrlRegs.GPBMUX1.bit GpioCtrlRegs.GPBDIR.bit.G GpioDataRegs.GPBSET.bit asm(" EDIS");	<- Enable EALLOW protected register access GPI032 = 0;<- GPI0032 GPI0 iPI032 = 1; <- GPI032 output .GPI032 = 1; <- GPI032 pin is set to 1 // Disable EALLOW protected register access
** DSP/BIOS에서 TINT2,DLOGIN SetDBGIER(IER   0x6000); *(volatile unsigned int *)0x	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
}	
* GPIO 설명 DSP는 특수 기능 레지스터에 접 사용 후 EALLOW Bit를 Clear한[	근 하기 위해서는 EALLOW Bit를 Set후 접근후 사용 한다. 다.
- GPIO 필터 설정 레지스터(입력, GPxCTRL : GPIO에 Qualific GPxQSELn : GPIO에 Qualific	시 몇 클럭 동안 유지시 입력핀을 유효 시킬 것 인지 사용할때) cation 를 사용할 때 CLK 종류 지정 cation 를 사용할 때 값 설정
- GPIO 설정 레지스터 GPxMUXn : 0-3값으로 GPIO GPxDIRn : GPIO핀을 입력 GPxPAUn : GPIO핀에 PULI	O 또는 특정 기능 핀으로 I/O핀을 정의 (0) 또는 출력(1) 으로 설정 _ UP 저항을 설정
- GPIO 데이터 레지스터 GPxDATn : GPIO 입출력 GPxSETn : 해당 비트가 GPxCLEARn : 해당 비트가 GPxTOGGLEn: GPIO 해당 핀	데이터 레지스터 1이면 GPIO 핀을 High로 유지 시킴 1이면 GPIO 핀을 Low로 유지 시킴 !을 TOGGLE 함.

### - CCS3.3 DSP/BIOS PRD 생성

\* PRD란 스케줄링 에서 Clock Manager의 기본 클럭을 사용하여 주기적인 시간 마다 발생되는 Time Task로서 이 루틴은 타이머 인터럽트의 일종이므로 소스 코딩시 정체되는 코드나 너무 긴 시간이 소요 되는 코딩은 자제 해 주시기 바랍니다.



1. Main\_Bios.c 를 open후 아래 소스코드를 입력 후 저장한다.

아래 함수에 있는 실행 코드는 C소스의 타이머 인터럽트와 유사 하므로 최대한 간단 명료 하게 코딩 하고 while() 문처럼 무한 루프에 빠지는 코드는 작성 하지 않는것이 원칙 이다 다른 스케줄러에 영향을 줌.

void lo\_Blink(void) <- PRD 스켈줄 함수 명 { GpioDataRegs.GPBTOGGLE.bit.GPIO32 = 1; <- GPIO32 토글

}

2. DSP/BIOS Config->\*.tcf 를 실행 한다.



3. 스케줄링 기본 스케줄 클럭을 확인 한다.



- 4. DSP/BIOS에 설정된 기본 클럭은 1ms 이다.
  - 이 클럭은 아래 PRD(Periodic Function Manager)에 기본 클럭(1 Tick)으로 사용 된다.

CLK - Clock Manag	er 등록 정보			
General				
Object Memory:	L4SARAM	•		
🔲 Continue to run	on sw breakpoint	(free run)		
💌 Enable CLK Ma	nager			
🔽 Use high resolu	tion time for intern	al timings		
Microseconds/Int:	1000.0000			
🔲 Directly configu	re on-chip timer re	gisters		
🗖 Fix TDDR				
TDDR Register:	2			
PRD Register:	49999			
Instructions/Int:	150000			
확인	취소	적용(4	0	도움말

5. 스케줄링 PRD(Periodic Function Manager) 에서 아래와 같이 PRD를 생성 한다.



\* PRD TASK 관리 명을 입력 한다



\* Task 관리자 생성 확인



\* 생성한 PRDTask에 사용자 환경을 설정 한다.(blink\_PRD0 선택후 오른쪽 버튼)



\* General 에서 기본 정보를 설정 한다.

blink_PRD0 등록	루 정보			
General				
comment:	<add comments="" here=""></add>			
period (ticks):	500	-		실행할 Tick수(1ms*500 = 500ms)
mode	continuous 💌			
function:	_lo_Blink	-		실제 실행할 함수명 앞에 _를 붙임.
arg0:	0x0000000			
arg1:	0x0000000			
period (ms):	32767.0			
 확인	취소	적용( <u>A</u> )	도움말	

\* Task Manager에서 생성 된 PRDTask를 확인후 \*.tcf 파일을 종료 합니다.

F28335_example_BIOS_flash.tcf         Estimated_Data Size: 1250 Est. Min. Stack Size (MAUs): 254         System         Firstrumentation         CLK - Clock Manager         CLK - Clock Manager         Firstrumentation	-				
Estimate Data Size: 1250 Est. Min. Stack Size (MAUs): 254 CRD - Periodic Function Manager object CLK - Clock Manager CLK - CLK - CLK - CLK - CLK - CLK -	F283	35_example_BIOS_flash.tcf			
<pre>Sostem Instrumentation Solution So</pre>	Estimate	d Data Size: 1250 Est. Min. Stack Size (MAUs): 254	PRD - Periodic Function Manager object	Textual Configuration Script - Line: 1	
<pre>bios.MEM.instance("ZONE7").len = 0x20000; bios.MEM.instance("ZONE7").createHap = 0; bios.MEM.instance("ZONE7").space = "code"; bios.MEM.instance("FLASH").destroy(); bios.MEM.create("FLASH ABCDEFEGH"); docs.MEM.create("FLASH ABCDEFEGH");</pre>		a bala size: 1250 Est. Min. Stack Size (MAUS): 254 istern strumentation cheduling CLK - Clock Manager PRD - Periodic Function Manager PRD - Periodic Function Manager FRD - Periodic Function Manager TSK - Task Manager TSK - Task Manager ↓ IDL - Idle Function Manager Synchronization nput/Output	Execution Order (First)	<pre>textualCongGradion Schpt Line 1 utils.loadPlatform ("ti.platforms.ezdsp28335"); /* The following DSP/BIOS Features are enabled. */ bios.enableRealTimeAnalysis(prog); bios.enableRealTimeAnalysis(prog); bios.enableTskManager(prog); bios.GEL.CLKIN = 30000; bios.GEL.CALLUSERINITFXN = 1; bios.GEL.CALLUSERINITFXN = 1; bios.GEL.MODIFYPLLCR2 = 0; bios.MEM.instance("SRAM").destroy(); bios.MEM.instance("ZONE7").comment = "External - 128Kw . bios.MEM.instance("ZONE7").eraetHeap = 0; bios.MEM.instance("ZONE7").eraetHeap = 0; bios.MEM.instance("FLASH").destroy(); bios.MEM.instance("FLASH").eftroy(); bios.MEM.instance("FLASH").eftroy(); bios.MEM.instance("ZONE7").eraetHeap = 0; bios.MEM.instance("FLASH").eftroy(); bios.MEM.instance("FLASH").eftroy();</pre>	•

Code Con	nposer Studio			x
⚠	변경 내용을 F28	335_example_BI	OS_flash.tcf에 저	장할까요?
	예攽	아니오( <u>N</u> )	취소	

# - 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 ▼ Open <u>D</u> ependent Projects When Loading Projects ■ Do <u>N</u> ot Scan Dependencies When Loading Projects ■ <u>A</u> uto-save Projects Before Build ▼ Open Project <u>W</u> indow 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	Tools
<sup>™</sup>	<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 <b>7</b>	
	<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	
- <b>*</b>	🖻 F	ЦX	e Ri	in al				F28x	x On-Chip F	lash Progr	ammer	<b>N</b> ?
							-	Data	Converter S	upport		· ·

\* 아래 CLOCK 설정 메뉴를 사용자에 맞게 설정 합니다.

<b>Clock Configuration</b>		×
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			
	50E03537E210,031			
파일 이름( <u>N</u> ):	FlashAPIInterface28335V2_10,out		<b>_</b>	
파일 형식( <u>T</u> ):	COFF Files (*,out)		▼ <sup>*</sup>	1소

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					
C:\CCStudio_v3,3\plugins\Flash28xx\Algorithms\28335\FlashAPIInterface28:					





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 <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 10</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 10</td>	00 10
Flash Programmer Settings		
•	▼   ↓	



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

6. Main\_Bios.c의 Io\_Blink() 에 break mode를 설정 합니다.



- F5 : Debuf->Run
- F9 : Debuf->Break Point Toggle