### \* CCS3.3 DSP/BIOS 환경 만들기

- 1. TI제공 예제 파일을 다운 받은후 압축을 작업 폴더에 풉니다. . spra958h.zip 예제프로그램
- 2. CCS3.3 폴더에 설치된 DSP BIOS버전을 확인후 최신버전(5.33.xx 이상)으로 업그레이드 한다. 업그레이드된 BIOS버전을 Code Composer Manager에 등록 한다.





3. Setup CCStudio v3.3을 실행 합니다.



4. 프로그램 및 디버거에 사용할 장치를 선택 한후 Add버튼을 클릭 합니다.

顰 Code Composer Studio S	Setup				
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp				<u> </u>	
System Configuration	Augilable Factory Ba	rde	Family	Platform	Endianness
	Available Factory Dua	ius	С28хх 🗾	xds510usb 💌	All 👤
🖳 My System	📷 F2801 XDS510USB	Emulator	C28xx	xds510usb	*
	📷 F2806 XDS510USB	Emulator	C28xx	xds510usb	*
	📭 F2808 XDS510US <mark>B</mark>	Emulator	C28xx	xds510usb	*
	📭 F2810 XDS510USB	Emulator	C28xx	xds510usb	*
	F2811 XDS510U3B	Emulator	C28xx	xds510usb	*
	F2812 XDS51005B	Emulator	C28xx	xds51Uusb,,,	*
	F28332 XDS5100SB	Emulator	C28xx	xds51Uusb,	*
	••••••••••••••••••••••••••••••••••••••	Emulator	C28xx	xds51Uusb	*
	F28335 XUS510USB	Emulator	C28xx	Xds51Uusb	*
	— - · · · · ·	📼 Custore Rearda I 🐔	Croote Reard		
	Factory Boards		Create Duaru	1	
Save & Quit Remove	< Add	< Add Multiple			
Drag a device driver to the left to a	dd a board to the syste	em,			

🐉 Code Composer Studio	Setup	
<u>File E</u> dit <u>V</u> iew <u>H</u> elp		
System Configuration	Available Pro	Driver Location
I My System I F28335 XDS510USB Emu I Cpu_0	★ TMS320F2400 ★ TMS320F2800 ★ BYPASS	C:₩CCStudio_v3,3₩drivers₩s, C:₩CCStudio_v3,3₩drivers₩s,
<	📷 Factory Boa	rds 📴 Custom Boards 🍖 Cr 📢
Save & Quit Remove	<< Add	<< Add Multiple
Select the system node to add a	new board to the s	system configuration,

5. My System에서 F283335 XDS510USB Emu를 선택후 Start Code Composer Studio를 실행 합니다.

🐌 Code Composer Studio	Setup		
<u>File E</u> dit <u>V</u> iew <u>H</u> elp		_	
<u>R</u> emove All Reventes Court Costiennetics	Ctrl+N	roc	Driver Location
Revert to Saved Configuration Save	Ctrl+O Ctrl+S	0F2800	C:₩CCStudio_v3,3₩drivers₩s
<u>I</u> mport, <u>E</u> xport,			
Start Code Composer Studio			
E <u>x</u> it	Alt+F4		
Save & Quit Remove	Fact	ory Boar	rds 🚾 Custom Boards 🌨 Cr. া

## \* CCS3.3 DSP/BIOS 예제 실행

1. Setup CCStudio v3.3 이나 CCSStudio3.3을 실행 합니다.



2. 아래와 같이 Project를 오픈 합니다.(Project->Open)



Project Open			? 🗙	
찾는 위치(!):	🗀 projects	👻 🕂 🔁 (	≝	
🚞 Debug				
F28335_examp	le_BIOS_flash.CS_			디 ^ 이에 다 시해 디 드
F28335_examp	le_BIOS_ram.pjt			프로젝트 프로젝트
F28335_examp	le_nonBIOS_flash.pjt			
F28335_examp	Ne_nonBIOS_ram.pJt			
	F28335 everaple BIOS flack pit		97(0)	비트 크리
파일 이름( <u>N</u> ),				
파일 영식( <u>1</u> );	Project Files (*,pjt)	<u> </u>	쥐소	
			도움말( <u>H</u> )	



3. 프로젝트 옵션 설정(Project->Build Options)



. 컴파일러 설정

Build Options for F28335_example_BIOS_flash.pjt (Debug)
General Compiler Linker DspBiosBuilder Link Order
-g -k -q -ss -pdsw225 -pds303 -al -as -fr"\$(Proj_dir)₩Debug" - fs"\$(Proj_dir)₩Debug" -ft"\$(Proj_dir)₩Debug" -fb"\$(Proj_dir)₩Debug" - ff"\$(Proj_dir)₩Debug" -i"\$(Proj_dir)" -i"\$(Proj_dir)₩,,₩include" -
Category: Hasic Advanced Feedback Files Assembly Parser Preprocessor Diagnostics Hasic Target Version: C28xx (-v28) Generate Debug Info: Full Symbolic Debug (-g) Opt Level: None Program Level Opt,: None
확인 취소 도움말

# . 링커 설정

Build Options f	or F28335_example_	BIOS_flash.pjt	(Debug)	? 🗙
General   Cor	mpiler Linker Dsp	BiosBuilder   L	ink Order	
-g -a -c -m o"\$(Proj_dir)	"\$(Proj_dir)₩Debug₩ ₩Debug₩F28335_ex	/F28335_exam ample_BIOS_fl	ole_BIOS_flash,m: ash,out" -w -x	ap" - 🔺
				· ·
Category:	Basic			
Basic	🔽 Suppress Bann	er (-q)		
Advanced	Output Module:	Abso	lute Executable (-	-a) 🔻
	Output Filename (-	-o): \$(Pro	j_dir)₩Debug₩F2	8335_example
	Map Filename (-m	i): \$(Pro	j_dir)₩Debug₩F2	8335_example
	Autoinit Model:	Run-	Time Autoinitializa	ation (-c) 💌
	Heap Size (-heap)	:		
	Far Heap Stack (-f	arheap):		
	Stack Size (-stack	):		
	Fill Value (-f):			
	Code Entry Point (	-e):		
		확인	취소	도움말

#### . DSP/BIOS 설정

Build Options for F28335_example_BIOS_flash.pjt (Debug)   ?     General   Compiler   Linker   DspBiosBuilder   Link Order     -Dconfig.importPath="\$(Install_dir)/bios_5_31_08/packages;" -v28   •   •	
Category: Basic Advanced Debug \$(Install_dir)/bios_5_31_08/packages;	— CCS3.3이 설치된 폴더에서 DSP/BIOS 폴더 위치를 지정
확인 취소 도움말	선택

4. JTAG 및 에뮬레이터를 연결 합니다.

<b>@</b> /	F2833	5 XDS	51 OUSE	Emula	tor/cp	ou_0 -	TMS32	OC28xx	- Code	e Ci
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>P</u> roject	<u>D</u> ebug	<u>G</u> EL	<u>O</u> ption	P <u>r</u> ofile	<u>T</u> ools	D <u>S</u> P/BIC	DS
眢	🖻 目	X	e C	<u>B</u> reak	points,					2
test	t, pjt		<b>_</b>	Asse Step	mbly/S Into	Source S	tepping	F11	۰,	\$
R.	66			Step . Step	<u>O</u> ver O <u>u</u> t			F10 Shift+F11		
እድ <sub>ር</sub> ପ୍ରତ୍ରପ୍ର		iles GEL Proje <b>d</b> ti	files ects est.pjt (C	<u>B</u> un Halt Anima Low I Run t Set P R <u>e</u> sta <u>G</u> o M Multip	ate Power o Curs C to Cu irt ain ole Ope	Run or ursor eration,.,		F5 Shift+F5 Alt+F5 Ctrl+Alt+F Ctrl+Alt+F10 Ctrl+Shift Ctrl+Shift Ctrl+Shift	-5 ++F10 ++F5	
****	-			Adva <u>i</u> Reset Reset Halt o	nced R t CPU t Emula in Resa	esets ator et		Ctrl+R Ctrl+Shift	► +R	
Ⅲ [詞				<u>C</u> onn Resto	ect ire <u>D</u> eb	ug State		Alt+C	•	

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

Customize 🛛 🔀	
Debug Properties   Directories   Color   Keyboard   Program/Project/Cl0   Control Window Dist	
	_ 체크
Disable All Breakpoints When Loading New Programs	
Project     I ✓ Open Dependent Projects When Loading Projects     □ Do Not Scan Dependencies When Loading Projects     □ Auto-save Projects Before Build     I ✓ Open Project Window On Startup	
CIO Maximum # of lines: 256 (may require restart)	
확인 취소 적용(△) 도움말	

#### 6. 컴파일 하기(Project->Rebuild All)





에러 확인

7. FLASH에 프로그램 하기

🏶 /I	2833	5 XDS	51 OUSE	) Emula	tor/c	0 - TMS32	DC28xx	- Code C	ompose	r Studio -	- Not
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>P</u> roject	<u>D</u> ebug	<u>G</u> EL	Option Profile	<u>T</u> ools	D <u>S</u> P/BIOS	<u>W</u> indow	<u>H</u> elp	
睂	🖻 🖬	I X	e Ri	in al			F28x	x On-Chip F	lash Progr	ammer	<b> </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			
	.62000042_10,000			
	Elach ADIInterface 28335\/2_10_out			71(0)
파일 혀신(T):				
ше с ¬\ <u>т</u> /-			<sup>-</sup>	1 <u></u>

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\pmplugins\Flash28xx\PAIgorithms\28335\FlashAPIInterface28.						



8. 프로그램을 로딩후 Debug 탭에서 Debug기능을 선택 실행 합니다.



- 9. 실행
  - Debug 창 에서 Run(F5)을 클릭 하면 바로 실행
  - Debug 창 에서 Go Main후 Run(F5)이나 Debug 메뉴 실행

휋 /F2	8335 XI	0\$510	USB Emula	tor	/cpu_(	0 - TN	AS320C2	8xx - C	ode Com	poser St	udio	o - (Main	BIOS.c]		
🔶 Eil	e <u>E</u> dit	<u>V</u> iew	<u>P</u> roject	D	ebug	<u>G</u> EL	<u>O</u> ption	P <u>r</u> ofile	<u>T</u> ools	D <u>S</u> P/BI	OS	<u>W</u> indow	<u>H</u> elp		
웥 🖬	ê 🔒	X 🖣			<u>B</u> reak	points					è.	"in "n C	a 🕼	۵ ا	?
F2833	660 C	ple_Bl	OS V Del CS V Del CS S CS S		Assen Step ( Step ( <u>Run</u> <u>Halt</u> <u>Anim</u> Run <u>F</u> Low F	nbly/S into Out Out ate ree	Run	pping	F11 F10 Shift+F1: F5 Shift+F5 Alt+F5 Ctrl+F5 Ctrl+Alt+ Ctrl+Alt+	1 .F5	- 11 11 10 32 32	• & ude "D ude "F <i>Globa</i> 6 AdcB 2 PwmD 2 PwmP	***** SP283 283355 <i>I Var</i> uf[AD uty; eriod	**** 3x_I _exa c_BL _	)ev amp /es JF_
<sup>→</sup> 0 (†) ズ	Occument   Operation   Operation		Set Po R <u>e</u> star <u>G</u> o M Multip	C to C t ain ple Op	Cursor Deration		Ctrl+Shif Ctrl+Shif Ctrl+Shif Ctrl+M	t+F10 t+F5		****** ction: cripti	main	**** () ain	e** Fu		
*		+ 	Source DSP2833x F28335_BI F28335_ex		Adva <u>r</u> Reset Reset Halt c	<mark>nced F</mark> CPU Emul on Res	Resets ator set		<b>Ctrl+R</b> Ctrl+Shif	► t+R	- - -	main(v <i>CFU I</i> nitSys	oid) nitia Ctrl( Ctrl(	<i>lize</i> );	•** 7 <i>t i</i>
ja K					Dis <u>c</u> o Resto	nnect re <u>D</u> el	oug State		Alt+C			nitWat nitGpi	chdog o();	();	
					<u>T</u> hrea <b>Real-t</b> Rude	d Levi <b>ime <u>N</u> Real-1</b>	el Debugg <mark>1</mark> ode time Mod	jing e			ei	f EXAM	PLE_F	LASH Zasl	ዘ ነና
				V	Flush	<u>P</u> ipeli	ne on Ha	lt		1 20		rtinn		⇔ <i>Ran</i>	 nF11

10. DSP BIOS Software Logic 및 Debug Event Timing(DSP/BIOS->Execution)





11. DSP BIOS Profile Routines w/o halting the CPU(DSP/BIOS->Statistics View)

/F28335 XDS510USB Emulator/cpu_0 - TMS320C28xx - Code Composer Studio - [Main_BIOS.c]									
♦ Eile Edit View Project Debug GEL O	ption P <u>r</u> ofile <u>1</u>	<u>T</u> ools	D <u>S</u> P/BIOS	<u>W</u> indow	<u>H</u> elp				
鼈 📽 🖬 👗 🏷 🍽 💼 🗠 🗠 🗌		·	<u>R</u> TA Con Execution	trol Panel n Graph	4				
F28335_example_BIOS - Debug	💽 🕸 🛗 🗄	₩ 👗	<u>S</u> tatistics	View					
🔊 🚳 🗋 🖩 🛤 🖬 🗖 🗖 🌮			<u>M</u> essage Kernel/O	Log bject View					
Image: Second secon	// DSP/	~BIO;	<u>H</u> ost Cha <u>C</u> PU Loa	annel Cont d Graph	rol Z				

Statistics View				×
STS	Count	Total	Max	Average
LedBlink_PRD PRD_swi ADC_swi TSK_idle IDL_busyObj	2811 351489 3,51461e+007 0 180055	0 1695259416 inst 19646159451 inst 0 inst -8,05477e+007	0 8073 inst 2934 inst -2147483648 inst -193	0 4823,08 558,99 0,00 -447,35

12. 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 Con	nposer Studio - [Main_BIOS.c]
♦ File Edit View Project Debug GEL Option Profile Tools	D <u>S</u> P/BIOS <u>W</u> indow <u>H</u> elp
🋍 🚅 🖬   X 🖻 💼   🗠 🗠	RTA Control Panel
F28335_example_BIOS 💌 Debug 💽 😻 🛗 🛃	Statistics View
	Message Log
	Kernel/Object View
(가) 🔮 Files	<u>CPU Load Graph</u> 3x_Dev
€ 128335.ael	examp
Message Log	×
Log Name: trace	-
3160 LedPrdCount = 3160 🖌	Led Blink() 함수에서 표시
3161 LedPrdCount = 3161	
3163 LedPrdCount = 3163	
3168 LedPrdCount = 3168	
3169 LedPrdCount = 3169 3170 LedPrdCount = 3170	E
3171 LedPrdCount = 3171	
3176 LedPrdCount = 3176	
3177 LeaPraLount = 3177 3178 LedPrdCount = 3178	
3179 LedPrdCount = 3179	-
P Contraction of the second seco	

13. 변수 확인

확인하고자 하는 변수를 지정 후 오른쪽 마우스를 클릭 Add Watch Window 창을 연다.

// Glob	al Variables	
Uintlb Adc Uint32 Pwm	<u>E</u> ditor	
Uint32 Pwm	Add to Watch Window	
	Quick Watch	
/********	Open Degument	***

Watch Window			Ē	×
Name	Value	Туре	Radix 🔺	]
🖃 🔗 AdcBuf	0x0000C30C	unsigned int[50]	hex	4
♦ [0]	4095	Uint16	unsigned	
🔶 [1]	4095	Uint16	unsigned	
	4095	Uint16	unsigned	
💊 [3]	4095	Uint16	unsigned	
	4095	Uint16	unsigned	
§ [5]	4095	Uint16	unsigned	1
[6] 🛕 [	4095	Hint16	uncianad 🛛 🗶	1
& Watch Locals	6 Watch 1			

## \* DSP/BIOS 기본

1. DSP/BIOS 설정 파일(\*.tcf) 설명

DSP/BIOS는 일반 C 프로그램 과 다른 점은 BIOS Configuration File(\*.tcf)이 추가 된다. 이 파일은 CCS에서 그레픽/TEXT 에디터 로 변경 할수 있으며 주 내용은 아래와 같다.

- System Setup Tools
- Real -Time Analysis Tools
- Real -Time Scheduler
- Synchronization
- Real-Time I/O
- \*.tcf 파일은 컴파일시 \*cfg.cmd, \*cfg.sxx, \*cfg.hxx, \*cfg\_c.c, \*cfg.h, \*.cdb 파일이 생성된다.



\* System->Global Setting(오른쪽 마우스 클릭 후 Properties선택) - 사용할 CPU 클럭 및 기본 정보를 설정 한다.

Global Settings 등록 정보			
282xx / 283xx 2802x General 280x	/ 2803x     2	2834×   81×	
Target Board Name:	c28xx		- CPU Type
Processor ID (PROCID):	0		✓ CPU 클럭 정의
Board Clock in KHz (Informational Only):	30000		
DSP Speed In MHz (CLKOUT):	150.0000		
Specify RTS library			
Run-Time Support Library:			
PLL Type	283xx	•	
Memory Model	LARGE	•	
🔽 Call User Init Function 🗲			체크후 수행할 함수명 앞에 _를 붙여 저이 하다
User Init Function:	_UserInit		* 시스템 리셋후 호출 되는 사용자가 자성하 하스 g ipt 00하스 초기보부에서
🔽 Enable Real Time Analysis			국경원 업무 C_III_00업무 조가무군에서 호출 된다.
Enable All TRC Trace Event Classes			도등 BIOS 소기와 임수 다 사용사 소기 필요 함수를 추가 할수 있다
확인 취소	적용( <u>A</u> )	도움말	

\* System->Memory Section Manager

(오른쪽 마우스 클릭 후 Properties선택, 또는 메모리 항목 추가시 Inser MEM 선택) - 프로그램 에서 사용 되는 메모리를 정의 한다.

MEM - Memory Section Manager 등록 정보						
General BIOS Data BIOS C	ode   Compiler Sectio	ons   Load Addres	5]			
🗖 Reuse Startup Code Space						
Argument Buffer Size:	0x0004					
Stack Size (MAUs):	0x0200					
🔲 No Dynamic Memory Heaps						
Segment For DSP/BIOS Objects:	L4SARAM	-				
Segment For malloc() / free():	L4SARAM	•				
 확인	취소	적용( <u>A</u> )	도움말			

\* System->Memory Section Manager->ADC\_CAL(메모리명) (오른쪽 마우스 클릭 후 Properties선택) - 각 메모리별 주소와 사이즈 Location 위치를 지정 한다.

ADC_CAL 등록 정보	1	
General		
comment:	tion in Reserved memory	
base:	0x380080	
len:	0x0009	
🔲 create a heap	in this memory	
heap size:	0x0200	
🔲 enter a user de	efined heap identifier label	
heap identifier lab	el; segment_name	
space	code	
확인	<b>취소</b> 적용( <u>A</u> )	도움말

- \* Instrumentation->LOG-Event Log Manager (오른쪽 마우스 클릭후 Inser LOG를 선택 항목을 추가)
- \* Instrumentation- >LOG-Event Log Manager >trace(오른쪽 마우스 클릭 후 Properties선택) - LOG\_printf()를 통해 CCS3.3 에서 메시지를 확인 할수 있다.

trace 등록 정보			X	
General				
comment:	Kadd comments her	e>		
bufseg:	L4SARAM	•		
buflen (words)	64	•		
logtype	circular	•		LOG_printf(&trace, ) 호출시 printf() 여격
datatype	printf	•		
format:	0x%x, 0x%x, 0x%x			
확인	취소	적용( <u>A</u> )	도움말	

- \* Scheduling->CLK-Clock Manager(오른쪽 마우스 클릭후 Inser CLK를 선택 항목을 추가)
- \* Scheduling->CLK-Clock Manager(오른쪽 마우스 클릭 후 Properties선택 ) - 스케쥴링에 사용 되는 기본 클럭을 설정 한다.

LK - Clock Manager 등록 정보					
General					
Object Memory:					
Continue to run on sw breakpoint (free run)					
🔽 Enable CLK Manager					
Use high resolution time for internal timings					
Microseconds/Int: 1000.0000	1000us->1ms				
Directly configure on-chip timer registers					
Fix TDDR					
TDDR Register: 2					
PRD Register: 49999					
Instructions/Int: 150000					
<u>확인</u> 취소 적용( <u>A</u> ) 도움말					

- \* Scheduling->PRD-Peroiodic Function Manager (오른쪽 마우스 클릭후 Inser PRD를 선택 항목을 추가)
- \* Scheduling->PRD-Peroiodic Function Manager->LedBlink\_PRD(오른쪽 마우스 클릭후 Properties선택) - LedBlink함수를 호출하는 시간을 정의 하고 있다.

(작성시 이 함수 안에서	1 오래머무르는 기능은	제외 시키길 권장 합니다.
:: 일반 C의 Timer 인터	럽트라 생각 하시면 됨	)

LedBlink_PRD {	등록 정보		X	
General				
comment:	Kadd comments here>			
period (ticks):	250 🔫			CLK->Clock Manager에서 설정한
mode	continuous 💌			시간(1ms) * 250 = 250ms 250Tick이라고 함
function:	_LedBlink 🔸			호출 되는 함수 명 앞에 _를 붙임
arg0:	0x0000000			
arg1:	0x0000000			
period (ms):	250.0			
확인	취소	적용( <u>A</u> )	도움말	

\* Scheduling->HWI->PIE Interrupts->PIE\_INT1\_6(오른쪽 마우스 클릭 후 Properties)

 시스템에서 사용되는 인터럽트 함수를 등록 한다.
특이한 사항은 Disspatecher항목이 있어 하드웨어 인터럽트에서 발생한 인터럽트를 바로 처리 하지 않고 SWI(소프트웨어 인터럽트) 인터럽트를 처리하는 함수(TASK)를 미리 생성해 그곳에서 인터럽트를 처리 하도록 하는 기능이 있다. 이때 HWI루틴에서는 SWI\_post() 를 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.

PIE_INT1_6	등록 정보			X	ſ	
General	Dispatcher					
comment:	defines function for t	he P				
function:	_ADCINT_ISR	-			호출 되는	: 함수 명 앞에 _를 붙임
monitor	Nothing	-				
addr:	0x0000000					
type	signed	$\overline{\mathbf{v}}$				
operation	STS_add(*addr)	-				
확인	취소		적용( <u>A</u> )	도움말		

PIE_INT1_6 등록 정보			X	
General Dispatcher	]			
✓ Use Dispatcher				세크 하지 않으면 HWI, 세크하면 SWI
Arg:	0x00000000			
Interrupt Mask IER0	self	•		
Interrupt Bit Mask IER:	0x0001			
확인	취소	적용( <u>A</u> )	도움말	

\* Scheduling->SWI

(오른쪽 마우스 클릭 후 Properties선택 또는 오른쪽 마우스 클릭후 Inser SWI를 선택 항목을 추가 할수 있다) \* Scheduling->SWI ->ADC\_SWI(오른쪽 마우스 클릭 후 Properties선택)

ADC_swi 등	록 정보		X	
General				
comment:	Kadd comments here>	[		
function:	_AdcSwi 🚽			_ 호출 되는 함수 명 앞에 _를 붙임
priority	2 🛨			- 호출 되는 함수 우선순위 레벌
mailbox:	0			
arg0:	0x0000000			
arg1:	0x0000000			
확인	취소	적용( <u>A</u> )	도움말	

- \* Scheduling->TASK(오른쪽 마우스 클릭후 Inser TASK를 선택 항목을 추가) - 하나의 별도 프로그램 이며, BIOS의 스케쥴러에 의해 관리되며 실행 된다.
- \* Synchronization-SEM(오른쪽 마우스 클릭후 Inser SEM을 선택 항목을 추가) - TASK 와 TASK간 동기나 자료를 전달하기 위해 사용.
- \* Synchronization-MBX(오른쪽 마우스 클릭후 Inser MBX을 선택 항목을 추가) - TASK 와 TASK간 자료를 전달하기 위해 사용.
- \* Synchronization-QUE(오른쪽 마우스 클릭후 Inser QUE을 선택 항목을 추가) - TASK 와 TASK간 자료를 전달하기 위해 사용.



