1. CCS3.3 DSP/BIOS semaphore를 이용한 COM1 시리얼 통신

기존의 예제 프로그램을 바탕으로 실제 현장에 적용 가능한 BIOS프로그램 을 시작 하겠습니다. 이 장에서 사용할 H/W는 F28335내부에 있는 UARTO 포트를 사용해 PC와 RS-232 통신을 프로세서간 동기를 맞추는데 사용하는 semaphore를 사용해 작성해 보겠습니다.

- Semaphore란

C 프로그램을 작성해 보신 분이라면 광역 변수의 0 과 1상태만 가지는 flag정도라 생각 하시면 됩니다. DSP/BIOS에서는 이 플레그 기능에 전 시간에 설명한 시간 지연(TSK_sleep()) 기능이 추가 되어 있습니다. 결론은 BIOS에서 각 TASK간 이벤트 동기와 시간 지연을 동시에 사용 할수 있게 만든 함수라 생각 하시면 됩니다.

- semaphore 설정 방법

CCS BIOS TOOL에서 1개 또는 여러 개의 semaphore를 사용자에 맞춰 생성 합니다.

- 변수명, 초기치..
- semaphore 사용법
 - SEM_post(&sem_com1_rx_eflg); 이벤트를 발생 시킬때 사용하는 함수 입니다. Item1: CCS BIOS에서 생성한 SEM 변수명 을 ()에 사용 합니다.
 - 2. SEM_pend(&sem_com1_rx_eflg,SYS_FOREVER); <- 리턴 있음 이벤트를 대기 할때 사용하는 함수 입니다. Item1: CCS BIOS에서 생성한 SEM 변수명 을 ()에 사용 합니다. Item2: 함수내 시간 지연 방법 설정 입니다. SYS_FOREVER : 이벤트가 SEM_post() 될때까지 영원히 기다림. 1 이상 숫자 : 1은 1TICK을 의미 하며 이 TICK동안 SEM_post()를 기다리다 탈출 합니다.
 - 위 경우는 리턴 되는 값을 확인하여 처리 합니다. 리턴: 0 이면 지정한 TICK동안 이벤트가 발생 하지 않았음.
 - 1 이면 지정한 TICK동안 이벤트가 발생함..
- TEST 환경(115200Bps,8 Bit, NON Parity, 1STOP)
 - PC : RS-232 PORT
 - DSP 보드 : UARTO RS-232
- 1. 디렉토리 구성
 - ..₩cmd : Linker 컴맨드 파일
 - ..DSP2833x_headers : Chip관련 헤더 파일 및 헤더용 Linker 컴맨드 파일
 - ..₩include : 사용자 인쿠르드 파일
 - : 사용자 프로젝트 파일 및 실행 파일(.HEX)
 - ..₩semsrc

..₩semprj

: 사용자 소스 파일

- CCS3.3 DSP/BIOS Semaphore 구성



1. Main_Bios.c 설명

```
void Uart1_RxSwi(void)
                                        <- SWI 스케줄 함수 명 :: 실제 UARTO 수신 처리
{
int i,rsr,len;
  rsr = SciaRegs.SCIRXST.all & 0xff;
                                        <- 수신 상태 레지스터
                                        <- 에러 검사
  if((rsr & 0x9c) != 0){
     SciaRegs.SCICTL1.bit.SWRESET=0;
                                        <- reset
     SciaRegs.SCICTL1.bit.SWRESET=1;
                                        <- non reset
  }
                                        <- 에러 없음.
  else{
     en = (SciaRegs.SCIFFRX.all >> 8) & 0x1f; <- 수신 FIFO에서 수신된 개수 읽음.
     for(i = 0; i < len; i++){
                                        <- 데이터중 최종 만 읽음.
       com1_rx_data = SciaRegs.SCIRXBUF.all;
       SEM_post(&sem_com1_rx_eflg);
                                        <- semaphore 생성(수신된 데이터 있음.)
     }
  }
  SciaRegs.SCIFFRX.bit.RXFFOVRCLR=1;
                                        <- Clear Overflow flag
                                        <- Clear Interrupt flag
  SciaRegs.SCIFFRX.bit.RXFFINTCLR=1;
}
```

2. 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(); InitKintf();	<- 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 ** memcpy(&secureRamFund &secureRamFun InitFlash();	cs_runstart, ics_loadstart,&secureRamFuncs_loadend - &secureRamFuncs_loadstart);
	com_init(_COM1,115200L,	NON_P,8,1);<- COM1 설정(DSP2833x_sci.c)
	asm("EALLOW"); GpioCtrlRegs.GPBMUX1.b GpioCtrlRegs.GPBDIR.bit.d GpioDataRegs.GPBSET.bi asm("EDIS");	<- Enable EALLOW protected register access it.GPIO32 = 0;<- GPIO032 GPIO GPIO32 = 1; <- GPIO32 output t.GPIO32 = 1; <- GPIO32 pin is set to 1 // Disable EALLOW protected register access
	** DSP/BIOS에서 TINT2,DLOGI SetDBGIER(IER 0x6000) *(volatile unsigned int *)0>	NT를 사용 하므로 BIOS사용 인터럽트 허가 ** ;
	** 아래 main()를 종료 하면 DSP	Y/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 (OM1 수신 처리)
	}	
	void lo_Blink(void){	<- PRD 스케줄 함수 명
	}	
	Void Uart1_RxSwi(void){	<- COM1_RX SWI 스케줄 함수 명
	}	
3	소스코드 설명 1. DefaultIsr_BIOS.c	
	void SCIRXINTA_ISR(void){	<- UART0 H/W(PIE9.1) 인터럽트를 바로 처리하지 않고 SWI로 변환하여 실행
	PieCtrlRegs.PIEACK.all =	PIEACK_GROUP9; <- Must acknowledge the PIE group
	SWI_post(&UART1_Rx_Sv	wi); <- SWI 로 변환
	}	

```
2. Main_Bios.c 설명
  void Uart1_RxSwi(void)
                      <- SWI 스케줄 함수 명 :: 실제 UARTO 수신 인터럽트 처리
  ł
  int i,rsr,len;
    rsr = SciaRegs.SCIRXST.all & 0xff;
                                        <- 수신 상태 레지스터
    if((rsr \& 0x9c) != 0)
                                        <- 에러 검사
       SciaRegs.SCICTL1.bit.SWRESET=0;
                                        <- reset
       SciaRegs.SCICTL1.bit.SWRESET=1;
                                        <- non reset
    }
    else{
                                        <- 에러 없음.
       en = (SciaRegs.SCIFFRX.all >> 8) & 0x1f; <- 수신 FIFO에서 수신된 개수 읽음.
      for(i = 0; i < len; i++)
                                        <- 데이터중 최종 만 읽음.
         com1_rx_data = SciaRegs.SCIRXBUF.all;
         SEM_post(&sem_com1_rx_eflg); <- semaphore 생성(수신된 데이터 있음.)
       }
    }
                                        <- Clear Overflow flag
    SciaRegs.SCIFFRX.bit.RXFFOVRCLR=1;
    SciaRegs.SCIFFRX.bit.RXFFINTCLR=1;
                                        <- Clear Interrupt flag
  }
                                        <- 사용자 TASK :: UART0 관리
  void task1_proc(void)
  {
    while(1) {
       SEM pend(&sem com1 rx eflg,SYS FOREVER);<- 1바이트가 수실될때 까지 무한정 기다림
       com1_putch(com1_rx_data);
                               <- 받은 데이터 송신(폴링 방식)
    }
  }
          ↓ TICK을 지정할 경우
  void task1_proc(void)
                                        <- 사용자 TASK :: UART0 관리
  {
    while(1) {
      if(SEM_pend(&sem_com1_rx_eflg,500)){ <- 1바이트가 수실될때 까지 500 TICK 기다린 후 처리(리턴참조)
          com1_putch(com1_rx_data); <- 받은 데이터 송신(폴링 방식)
       }
    }
  }
```

* CCS3.3 DSP/BIOS 구성 및 설명

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



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

휋 /F28335 XDS	10USB Emulator,	/cpu_0 - TMS32	0C28xx - Code (Composer Studio - Not Connected
<u>F</u> ile <u>E</u> dit <u>V</u> iew	Project Debug	<u>G</u> EL <u>O</u> ption	P <u>r</u> ofile <u>T</u> ools	D <u>S</u> P/BIOS <u>W</u> indow <u>H</u> elp
12 🚅 🗐 🐰	<u>N</u> ew			TT 🖌 🏹 🐔 🐂 😘 🎒 🔗 📢
-	Open			
	Us <u>e</u> External	Makefile	1	i 👗 🖑 🖑 🛡 🔑
	Export to Ma	kefile		I
Ka 00 🔲	Add Files to I	Project		

Project Open			? 🗙	
찾는 위치([):	🚞 sem_prj	- 🕈 🖻	*	
F28335_examp	ole_BIOS_flash.CS_			FLASH에서 실행되는
120000_examp				프로젝트
파일 이름(<u>N</u>):	F28335_example_BIOS_flash.pjt		열기(<u>0</u>) ┥	버튼 클릭
파일 형식(<u>T</u>):	Project Files (*,pjt)	•	취소	
			도움말(<u>H</u>)	

🔮 Files	
🗄 🖷 🧰 GEL files	
🗄 📲 Projects	
🗄 🚵 F28335_example_BIOS_flash.pjt (Debug)	
Dependent Projects	
Documents	
🖅 💼 DSP/BIOS Config 🛛 🛶 🕡	DSP/BIOS Config 파일
🖅 💼 Generated Files	
🗉 🗀 Include	
💼 Libraries	
	Watch-dog disable후 C를 실행 할때 사용, 필요시
CodeStartBranch.asm	project->build option에서 code entry piont에서 등록하여 사용
	인터럽트 서비스 루틴
	DelayUs() 함수 지원
	전역데이터 및 데이터 섹션 정의
🔝 DSP2833x_sci.c 🛶	통신 프로그램 소스
	Flash Memory 관련 지원 및 초기화
🔛 Gpio.c 🔸	CPU I/O핀 초기화
🔝 Main_BIOS.c <	Main() 프로그램
Passwords.asm 🖌	Flash Passwords 관련
PieCtrl_BIOS.c 🔸	CPU 인터럽트 초기화
🔝 SetDBGIER.asm 🖌	BIOS 인터럽트 지원
SysCtrl.c	CPU 클럭 설정
🔝 Watchdog.c <	Watch-Dog 관련 초기화
🔛 🔝 Xintf.c 🛛 🔸	외부 버스 초기화(wait 설정)
DSP2833x_Headers_BIOS.cmd	DSP/BIOS Config에서 컴파일시 생성
F28335_BIOS_flash.cmd	기본 CMD 파일 정의
🔤 🔛 F28335_example_BIOS_flashcfg.cmd	· DSP/BIOS Config에서 컴파일시 생성

4. DSP/BIOS Config->*.tcf 를 실행 한다. TASK생성, LOG_printf 생성법은 기존 자료 참조 하세요.



5. System->MEM->L5,L6,L7SARAM을 삭제 한다.(마우스 고정후 삭제 키) BIOS 사용 메모리 확장



6. System->MEM->L4SARAM 내용 수정



L4SARAM 등록 정보			
General			
comment:	On-Chip RAM Memory		
base:	0x00c000		
len:	0x4000		메모리 크기 수정(0x4000)
🔽 create a heap in t	this memory		L4-L/베모리 영역 모두 시장
heap size:	0x0200		
🔲 enter a user defin	ed heap identifier label		
heap identifier label:	segment_name		
space	data 🗨		
확인	쥐소 석용(<u>A</u>)	도움말	
			•

7. 하드웨어 인터럽트를 SWI로 변환(Scheduling ->HWI->PIE INTERRUPTS->PIE_INT9.1

🛄 PIE_IN	IT8_8	
····· 🕛, PIE_IA		오른쪽 마우스(PIE_INT9.1)
🕛, PIE_I	What's This?	
🕛 PIE_I	Undo	
🛄 PIE_I -		
🛄 PIE_I	Cut	
🛄 PIE_I	Сору	
🕛, PIE_I	Paste	
🕛, PIE_I	Incert Object	
🛄 PIE_I .	Insert Object	
🕛, PIE_I	Delete	
····· 🛄 PIE_I	Rename	
🖐 PIE_I '	- Deservativé values véseus	
🕛, PIE_I	 Property/value view 	
🕛, PIE_I	Properties 🚽	- 서택(왼쪽 마우스)
🕛, PIE_I		
🕛, PIE_I	Show Dependency	
dh ears an	etta a la	

* 하드웨어 인터럽트 등록(HWI)

PIE_INT9_1 5	등록 정보			
General	Dispatcher			
comment:	defines function for the P			
function:				호출 되는 함수 명 앞에 _를 붙임
monitor	Nothing			
addr:	0x0000000			
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
Se Dispatcher			
Arg:	0x0000000		
Interrupt Mask IER0	self		
Interrupt Bit Mask IER:	0x0001		
확인	취소 적용(<u>A</u>)	도움말	

8. 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 Configuration UART1_Rx_Swil	입력	후 확인
ОК	<u> </u>	

* 생성한 SWI에 사용자 환경을 설정 한다.(UART1_Rx_Swi 선택후 오른쪽 버튼)



* General 에서 기본 정보를 설정 한다.

UART1_Rx_S	wi 등록 정보			
General				
comment:	com1_rx_int			주석
function:	_Uart1_RxSwi 🛛 🚽			호출 되는 함수 명 앞에 _를 붙임
priority	2 🗲			호출 되는 함수 우선순위 레벌(1(저)-15(고))
mailbox:	0			
arg0:	0x0000000			
arg1:	0x00000000			
확인	취소	적용(<u>A</u>)	도움말	

9. Semaphore 등록(Synchronization->SEM semaphore Manager)



* semaphore 명을 입력 한다.(프로그램 에서 호출 변수가 됨.)

Insert Object	
Please enter a name for the C	onfiguration
sem_com1_rx_eflg	입력 후 확인
OK	

* 생성한 SEM에 사용자 환경을 설정 한다.(sem_com1_rx_eflg 선택후 오른쪽 버튼)

System Scheduling Scheduling Synchronizat SEM - Sem	on ion maphore Manager	오른쪽 마우스
MBX - N	What's This?	
	Undo	
	Cut Copy Paste	
	Delete Rename	
	Property/value view Properties	서태(외쪼 마으스)
	Show Dependency	

* General 에서 기본 정보를 설정 한다.

sem_com1_ r x_eflg 등록	정보		
General			
comment:	com1_rx_prod		주석
Initial semaphore count:	0		—Semaphore 초기치
확인	취소 적용(<u>A</u>)	도움말	

* Scheduling SWI에서 생성된 UART1_Rx_Swi 를 확인후 *.tcf 파일을 종료 합니다.

F28335_example_BIOS_flash.tcf *				_ D X
Estimated Data Size: 1278 Est. Min. Stack Size (MAUs): 323	UART1_Rx_Swi prope	rties	Textual Configuration Script - Line: 1	
Gystem Gystem	Property comment function priority mailbox arg0 arg1	Value com1_rx_int _Uart1_RxSwi 2 0 0x00000000 0x00000000	<pre>utils.loadPlatform("ti.platforms.ezdsp28335"); /* The following DSP/BIOS Features are enabled. bios.enableRealTimeAnalysis(prog); bios.enableTskManager(prog); bios.GBL.CLKIN = 30000; bios.GBL.CLKIN = 30000; bios.GBL.CLKIN = prog.extern("UserInit"); bios.GBL.MODIFYPLICR2 = 0; bios.MEM.NOMEMORYHEAPS = 0; bios.MEM.NOMEMORYHEAPS = 0; bios.MEM.instance("SRAM").destroy(); bios.MEM.create("20NE7").base = 0x20000; bios.MEM.instance("20NE7").len = 0x20000; bios.MEM.instance("20NE7").len = 0x20000;</pre>	*/ E



- CCS3.3 DSP/BIOS 실행

1. JTAG 및 에뮬레이터를 연결 합니다.

😻 /F28335 XDS510USB	Emulator/cpu_0 -	TMS320C28xx - Co	de C
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject	<u>Debug</u> EL <u>Option</u>	P <u>r</u> ofile <u>T</u> ools D <u>S</u> P/	BIOS
쒑 🚅 🔲 🐰 🖻 💼 📗	<u>B</u> reakpoints		124
[test,pjt[Assembly/Source S Step Into Step <u>O</u> ver	tepping F11 F10	•
?? Files ?? GEL files	Step O <u>u</u> t <u>R</u> un <u>H</u> alt <u>A</u> nimate Run <u>F</u> ree <u>L</u> ow Power Run Run to Cursor Set PC to Cursor R <u>e</u> start <u>G</u> o Main Multiple Operation,	Shift+F11 F5 Shift+F5 Alt+F5 Ctrl+F5 Ctrl+Alt+F5 Ctrl+Alt+F5 Ctrl+Shift+F10 Ctrl+Shift+F5 Ctrl+Shift+F5 Ctrl+M	
 ★ ★	Adva <u>n</u> ced Resets Reset CPU Reset Emulator Halt on Reset	ا Ctrl+R Ctrl+Shift+R	
III IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	<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 Program Load Perform verification during Program Load	_ 체크
 ✓ Open Dependent Projects When Loading Projects ☐ Do Not Scan Dependencies When Loading Projects ☐ Auto-save Projects Before Build ✓ 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

@ /	F2833	5 XDS	51 OUSE	8 Emula	tor/c	ри <u>0</u> – Тм	15320)C28xx	- 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 P	rofile	<u>T</u> ools	D <u>S</u> P/BIOS	<u>W</u> indow	<u>H</u> elp	
睝	🖻 🗖	IX	h R	in al				F28x	x On-Chip F	lash Progr	ammer	 ?
								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 ce28335V2_10, out <mark><</mark>			
파악 이르(N):	FlashAPIInterface28335V2_10_out		¥	37(0)
파일 형식(<u>T</u>):	COFF Files (*,out)		_	취소

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₩FlashAF	Plinterface28: Browse



Initialized RAM Warning	
Warning: Because this program con initialized RAM data it may run succ under Code Composer Studio but no If your Flash program requires initial	tains cessfully ot as a lized data in
Do not show this dialog again.	000 00 OK 최크후 확인

On-Chip Flash Programmer	i de la companya de l	같음
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 5 FFFF Key 4 FFFF Key 2 FFFF Key 1 FFFF Key 0 FFFF	Erase Sector Selection Image: Sector A: (338000-33FFF) Image: Sector F: (310000-317FFF) Image: Sector B: (330000-337FFF) Image: Sector G: (308000-30FFFF) Image: Sector C: (328000-32FFF) Image: Sector F: (300000-307FFF) Image: Sector D: (320000-327FFF) Image: Sector F: (300000-307FFF) Image: Sector D: (320000-327FFF) Image: Sector F: (300000-307FFF) Image: Sector D: (320000-327FFF) Image: Sector F: (300000-309FFF) Image: Sector D: (320000-317FFF) Image: Sector F: (300000-309FFF) Image: Sector D: (320000-317FFF) Image: Sector F: (30000-309FFF) Image: Sector D: (320000-309FFF) Image: Sector F: (30000-309FFF) Image: Sector D	10
Unlock Lock Program Password Flash Programmer Settings	OTP Wait State: 31 OTP: C Load RAM Only Execute Operation Help	

- /F28335 XDS510USB Emulator/cpu_0 TMS320C28xx Code Composer Stu <u>File Edit View Project Debug</u> GEL Option Profile Tools DSP/BIOS Breakpoints... 🎦 🚅 🔚 | X 🖻 💼 | 🗠 Assembly/Source Stepping ۲ F28335_example_BIOS - Del Step Into F11 📋 🗐 🔜 🗐 🖸 20 66^ Step Over F10 Step Out Shift+F11 **?**} 👰 Files 실행 <u>R</u>un F5 🛓 🔄 GEL files **P** Shift+F5 <u>H</u>alt 628335.gel {} Alt+F5 <u>A</u>nimate 🚊 🛅 Projects {} Ctrl+F5 Run Eree 🖻 📸 F28335_exam **9**, Low Power Run Ctrl+Alt+F5 📄 Dependent Run to Cursor Ctrl+F10 🛅 Document Set PC to Cursor Ctrl+Shift+F10 →{} 🗄 🚞 DSP/BIOS 🗄 🧰 Generated Restart Ctrl+Shift+F5 {[‡]} 🕂 🧰 Include <u>G</u>o Main Ctrl+M 🚽 _ Main() 함수 위치 로 이동 Ž 📄 Libraries Multiple Operation... 滂 🗄 💼 Source Advanced Resets Þ 🗄 DSP2833x ***** Reset CPU Ctrl+R 📩 F28335_BI Reset Emulator Ctrl+Shift+R F28335_ex öx Halt on Reset Disconnect Alt+C 阁 Restore Debug State 5 Ihread Level Debugging ٦ Real-time Mode Rude Real-time Mode ✓ Flush Pipeline on Halt
- 5. 프로그램을 로딩후 Debug 탭에서 Go Main -> RUN 기능을 실행 합니다.

6. BIOS 디버그 방법은 기존 BIOS 자료 를 참고 하세요.

* TEST 프로그램 실행

1. 하이퍼 터미널을 시작 함니다.



연결 설명

2. 하이퍼 터미널에서 통신 포트를 설정함니다.

💐 새 연결	
연결에 대한 이름을 입력하고 아이콘을 선택하십시오. 이름(<u>N</u>): 115200	
아이콘():	
 확인 취소	
연결 대상	
400 115200	
전화 걸 번호에 대해 자세히 입력하십시오.	
국가/지역(<u>C</u>): 대한민국 (82)	
지역 번호(E): 02	
전화 번호(<u>P</u>):	
연결에 사용할 모뎀(<u>N</u>): COM1	
· · · · · · · · · · · · · · · · · · ·	
독근 위또	
СОМ1 등로 저비 ? 조직	
포트 설정	
비트/초(B): 115200	보레이트
데이터 비트(D): 🔋	데이터 비트는 반드시 재 설정(하이퍼 에러)
패리티(P): 어우 🔍	
정지 비트(S): 1	
흐름 제어(<u>F</u>): 없음	
기본값 복원(<u>R</u>)	

? X

3. 하이 터미널이 실행되면 아무 키나 누르면 반송 됩니다.

