

RealDSP DataMon v1.95

프로그램 메뉴얼

(TMS320 LF240xA & F280x/281x/282x/283x & Piccolo F2802x/F2803x)



리얼시스

TEL : 031-342-3000

FAX : 031-343-0003

주소 : 경기도 안양시 동안구 호계동 안양IT밸리 504호

< 알 립 >

- 본 제품의 사용설명서 및 운용 프로그램은 제품의 성능 향상을 위하여 통보 없이 내용이 변경 될 수 있습니다.
- 본 제품의 하드웨어, 소프트웨어 및 관련 자료의 무단 복제, 수정을 금합니다.
- 본 제품의 무상 보증기간은 제품 구입 일로부터 1년으로 합니다.(단, 사용자의 취급 부주의 등으로 생긴 고장은 유상 수리 합니다.)
- 본 제품과 관련하여 사용자의 부주의로 인한 손실에 대하여 리얼시스는 책임을 지지 않습니다.

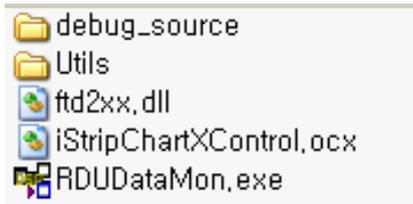
본 제품을 구입하신 후 사용함은 위의 알림에 동의함으로 간주 합니다.

Copyright © 2011 by 리얼시스

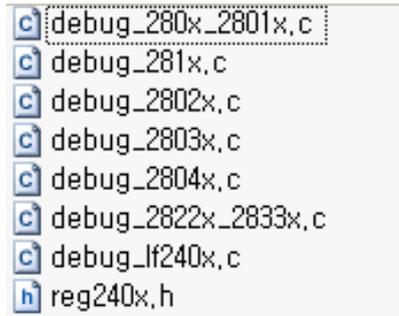
RealDSP DataMon Update History

< Ver1.95 >

- Piccolo F2802x(28021/28022/28023/28026/28027) MCU 다운로드 기능 추가
- Piccolo F2803x(28030/28031/28032/28033/28034/28035) MCU 다운로드 기능 추가
- 데이터 모니터링 관련 Firmware 소스를 RealDSP-UT 설치 파일에 포함 시킴



[RealDSP-UT 설치 폴더 내용]



[debug_source 폴더 내용]

RealDSP-UT의 변수 데이터 모니터링 방식은 debug_source 폴더에 있는 사용자 타겟 DSP 타입과 부합되는 시리즈 소스 파일을 사용자 프로그램 프로젝트에 같이 포함시킨 후 Compile & Link 한 COFF 파일을 타겟 보드에 다운로드 한 경우에 동작합니다.

< Ver1.8 >

- DSP28xxx 시리즈의 RAM Download 속도 개선
- RealDSP_UTA_Type 제품 지원을 위한 수정

< Ver1.7 >

- 지원 MCU 리스트에서 TMS320C28232/28234/28235/28332/28334/28335 추가
- POD Firmware 버전이 1.6이상에서만 지원함.

< Ver1.6 >

- 지원 MCU 리스트에서 TMS320C28015/28016/28044 and TMX320F28335 추가
- POD Firmware 버전이 1.6이상에서만 지원함.

< Ver1.5 >

- 지원 MCU 리스트에서 TMS320LF240x 삭제
현재 TI에서는 TMS320LF240x(Old Version) MCU가 더 이상 판매하지 않고 있으며 대신 TMS320LF240xA를 판매하므로 이번 버전에서 삭제하였음.
- 지원 MCU 리스트에 TMS320LF2401A 추가
TMS320LF2401A의 Flash API v1.3 사용

- TMS320LF240xA Flash API Upgrade
TMS320LF2402A/2403A/2406A/2407A의 Flash API를 v1.3으로 Upgrade
- TMS320F2810/2811/2812 Flash API Upgrade
TMS320F2810/2811/2812의 Flash API를 v2.10로 Upgrade
- TMS320F2801/2806/2808 Flash API Upgrade
TMS320F2801/2906/2808의 Flash API를 v3.02로 Upgrade
- 지원 MCU 리스트에 TMS320F2802/2809 추가
TMS320F2802 Flash API v3.02와 TMS320F2809 Flash API v1.0 사용
- TMS320F2801/2802/2806의 CSM Unlock 기능 개선
기존 RealDSP-UT v1.42 이하 버전의 경우 TMS320F2801/2802/2806 디바이스가 CSM Lock이 걸린 경우 Unlock기능이 되지 않아 문제가 되었는데 이번 버전에서 이 문제를 해결함. (Firmware 버전을 v1.5로 Upgrade 요망)
- TMS320280x/281x의 Chip 버전별 유의 사항(2006.08.28 테스트 결과)



위의 그림과 같이(CA버전) TI의 TMS320F28xx Chip에는 하단부에 고유 버전 정보가 있습니다. 그런데 최근 테스트 결과 Chip 버전이 CE버전 이하인 경우 RealDSP-UT 28x 윈도우 프로그램 버전이 v1.42이하에서는 Flash Writing이 잘 동작하나 Chip 버전이 CG버전 이상인 경우에는 Flash Writing이 잘 되지 않습니다. 이 경우 RealDSP-UT 28x 윈도우 프로그램 버전이 v1.50이상으로 Upgrade하셔야 합니다.

[RealDSP-UT 28x POD Firmware 수정 사항]

- TMS320LF240xA의 Flash API Code 다운로드시 에러 코드 수정
TMS320LF240xA의 Flash API Code 다운로드 시 각 단계별로 에러 상태 코드가 오는데 이번 Firmware에서는 이 에러 코드를 판별하여 에러 유무를 발생 시킴.
- TMS320F2801/2802/2806의 CSM Unlock 기능 개선을 위해 수정
TMS320F2801/2802/2806의 CSM Unlock 기능을 구현하고자 일부 기능을 변경하였음. 따라서 위의 MCU를 사용하시는 분들께서는 RealDSP-UT 28x POD의 Firmware를 체크하여 V1.5 이상인지 확인해 주시기 바랍니다.

< Ver1.4 >

- TMS320F280x(2801/2806/2808 – Flash API Ver3.0 사용) MCU 지원 추가
이 칩은 2005.10월에 한국에 출시 될 예정입니다.

< 기능상의 제한 사항 >

테스트 결과 CSM Lock이 걸린 TMS320F2801/2806의 경우 SCI-A 부트시 Flash API Algorithm Code를 Unsecured 0-wait internal RAM 영역에 다운로드해야하는데 다른 소자 (TMS320F281x 와 TMS320F2808)들과는 달리 H0 SRAM 영역이 없고 오직 M0 SRAM 영역(M1 영역은 부트 프로그램에서 스택으로 사용)만 사용할 수 있어 Flash API Algorithm Code를 다운로드 할 수 없어 CSM Lock을 풀 수 없습니다.

따라서 사용자께서는 개발시 CSM Lock을 걸지 말고 사용하시고 양산 시에만 CSM Lock을 사용하시길 바랍니다.

< Ver1.3 >

- RealDSP-UT POD에서 USB가 아닌 Serial 포트에 데이터 모니터링 시 자주 통신 에러가 발생하는 부분 수정 < RealDSP-UT Firmware 1.3으로 수정 >
- TMS320LF240x/A에서 COFF 파일 로드 시 데이터 RAM에 할당된 사용자 정의 섹션을 프로그램 Flash 영역에 덮어 씌우는 현상 수정
- TMS320LF240x/A의 CSM Unlocking 기능 보장
Lock이 걸린 Device에 대한 Unlocking 기능 추가
- Memory/Time Graph에서 최대 4개까지 선택한 변수에 대한 독립적인 그래프 그리기 지원
- Memory/Time Graph에서 최대 4개까지 선택한 변수에 대한 모니터링 데이터를 텍스트 파일로 저장할 수 있는 기능 추가
- TMX320F280x(2801/2806/2808 – Flash API Ver2.0 사용) MCU 지원 추가
이 칩은 2005.03월에 한국에 출시되었으며, 현재는 TMX C 버전만이 있습니다. 추후 TMP와 TMS 버전이 나오면 테스트 후 바로 자료실에 수정한 프로그램을 올리겠습니다.

< Ver1.24 이전 >

- USB 다운로드 속도 개선
- 예제 소스 중 Startup 소스 수정
- Flash Erasing 대화상자가 화면에서 자동으로 사라지지 않는 현상 수정
- 사용자 프로그램 다운로드 중 취소할 수 기능 추가
- TMS320F281x의 CSM Locking/Unlock 관련 기능 개선
- COFF 파일 처리에 있어 RAM function과 같이 LOAD와 RUN의 위치가 다른 섹션 처리 보장
- 예제 소스 중 TMS320F281x의 Internal Flash ROM 초기화 부분 수정

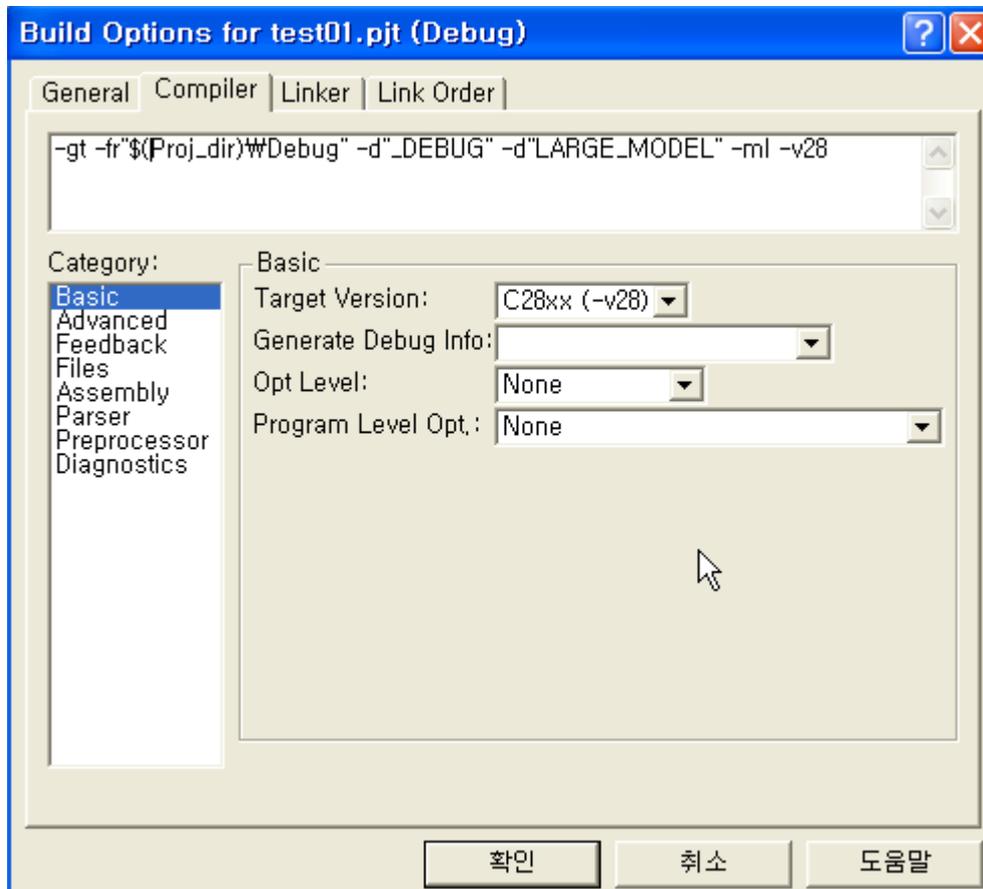
CCS v3.1 이상 및 C2000 Compiler v4.1.0 이상을 사용하는 분께

[현상]

CCS v2.2로 만든 COFF 파일을 RealDSP-UT for 28x 운용 윈도우 프로그램인 DataMon 프로그램에서 읽으면 debug symbol 정보 중 변수 타입 정보가 제대로 표시되는데 반해 CCS v3.1 또는 C2000 Compiler v4.1.0 이상으로 만든 COFF 파일에서는 debug symbol 정보 중 변수 타입 정보가 모두 integer 타입으로만 표시되는 현상이 있음.

[해결 방안]

CCS v3.1을 사용하시는 고객님들께서는 프로젝트 Build Options ->Compiler의 Basic -> Generate Debug Info 옵션을 No Debug로 하시고 직접 옵션 입력 창에 -gt 또는 --symdebug:coff 라고 입력하시면 위와 같은 현상이 해결될 것입니다.



그리고 DOS형 C2000 Compiler v4.1.0 이상 프로그램으로 batch 파일로 작성하여 개발하시는 분들은 make 파일의 컴파일 옵션에서 -g 옵션대신 -gt 또는 --symdebug:coff라고 입력하시길 바랍니다.

1. RealDSP DataMon 프로그램 및 USB POD 소개

RealDSP DataMon 프로그램은 TMS320LF240xA 과 TMS320F28x의 SCI 부트 기능을 사용하여 내부 플래시 롬(OTP ROM 또는 RAM 영역 for TMS320F28x)에 사용자 프로그램을 Writing 하며, 이 SCI 통신 포트를 사용하여 심볼 변수 및 주소의 메모리 내용을 모니터링 또는 변경이 가능합니다. 더불어 다운로드 속도를 향상시키기 위해 USB Interface를 채택하였습니다. 시스템을 개발하는 도중 개발 장비(XDS510 JTAG Emulator)가 고장 나거나 A/S를 요청하고 몇 개월 기다리는 경우에 개발자는 매우 답답할 수 있습니다. RealDSP DataMon 프로그램 및 USB POD 장치는 이런 개발자들에게 저가격의 편리한 개발 환경을 제공합니다.

2. RealDSP DataMon 프로그램 특징

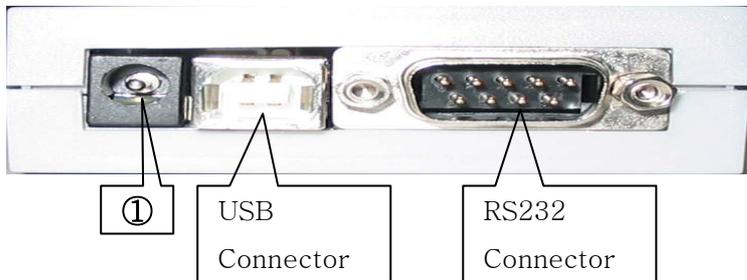
- A. 변수 및 주소 데이터 값 보기 및 변경 가능
- B. 사용자 도구 등록 및 실행 가능
- C. COFF 파일을 로드 하여 사용자 프로그램을 DSP 내부 플래시 롬(OTP ROM 또는 RAM 영역 for TMS320F28x)에 Download 하는 기능
- D. PC와 RS232 및 USB Interface 통신 인터페이스
- E. 변수 및 메모리 내용 그래프 표시 기능
- F. 전기적 절연 지원(절연형 DC/DC, Photo Coupler 사용)
- G. 전원 공급 : USB(5V) 또는 Target Board VCC(5V)
- H. FLASH API Ver2.10 사용(for TMS320F2810/2811/2812)
- I. FLASH API Ver3.02 사용(for TMS320F2801/2802/2806/2808/2809)
- J. FLASH API Ver1.3 사용(for TMS320LF2401A/2402A/2403A/2406A/2407A)
- K. FLASH API Ver1.0 사용(for TMS320F28015/28016/28044)
- L. FLAsH API Ver2.10 사용(for TMS320F28232/28234/28235)
- M. FLASH API Ver2.10 사용(for TMS320F28332/28334/28335)
- N. FLASH API Ver2.00 사용(for TMS320F28021/28022/28023/28026/28027)
- O. FLASH API Ver1.00 사용(for TMS320F28030/28031/28032/28033/28034/28035)
- P. Windows 32bit 98/Me/2K/XP/Vista/7 지원

3. USB POD 설치

1. USB POD 하드웨어 구성 및 설치

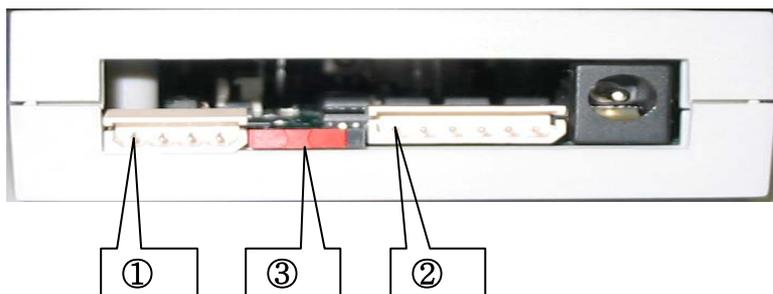


< PC측 연결부 - RS232 & USB Interface 지원 >

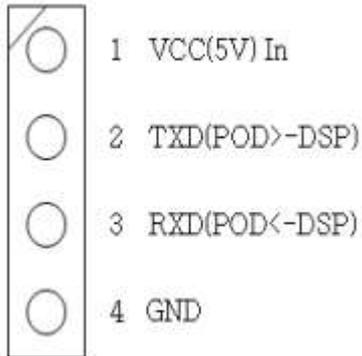


① : 5V 외부 전원 입력 단자로서 사용자 PC의 USB 포트의 전원이 불안정하거나, RS232 포트만을 사용하고자 할 때 USB POD에 5V 전원을 공급합니다.

< DSP측 연결부 - SCI Interface > - 개선된 모델에서는 3번 핀이 없음



①측 Connection 정보(232 Level)



②측 Connection 정보(TTL Level)



③ RS232 또는 TTL 신호 Level 선택 스위치

사용자의 Target Board와 통신할 신호 Level에 맞춰 사용하시길 바랍니다.

일부 RealDSP-UT에서는 이 선택 스위치가 없는 경우도 있습니다.

이는 개선된 모델로서 내부에서 자동으로 선택되도록 하였습니다.

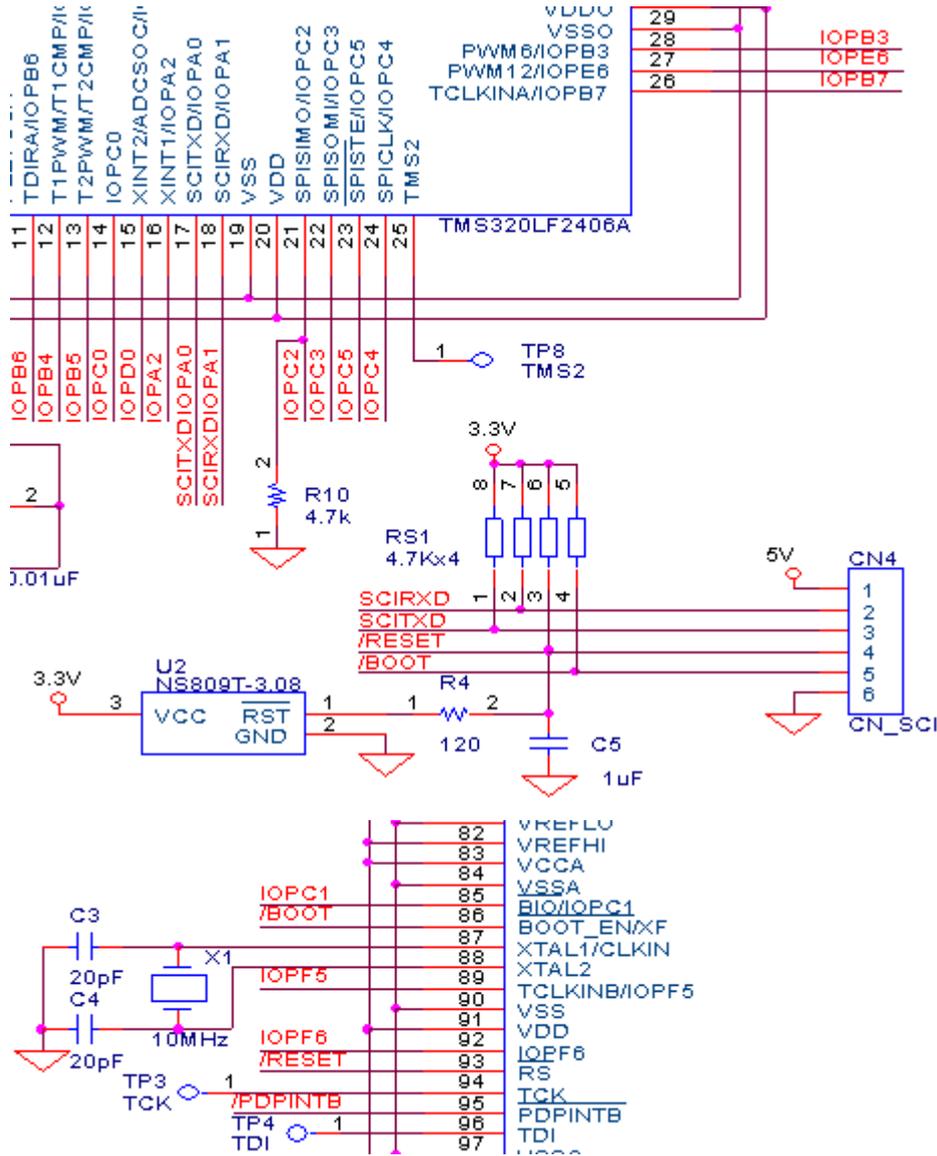
➤ TTL Level 인 경우(Default)

③번 스위치를 TTL Level(왼쪽으로 밀었을 경우)로 놓고 ②번 Connector에 제공되는 Cable로 연결하십시오.

➤ RS232 Level 인 경우

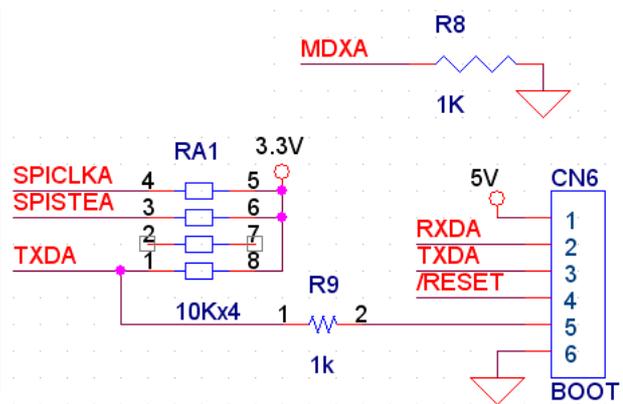
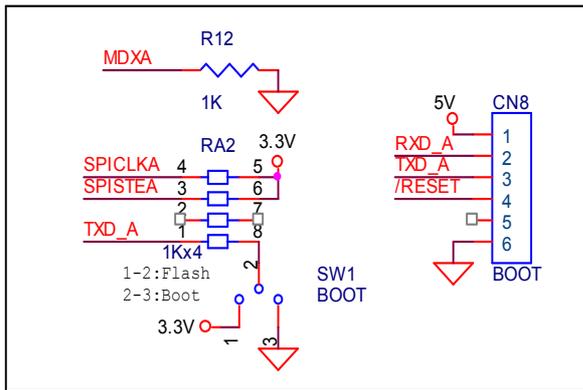
③번 스위치를 RS232 Level(오른쪽으로 밀었을 경우)로 놓고 ②번 Connector에서 /RESET 과 /BOOT Pin을 그리고 ①번 Connector의 4개 Pin을 묶어 사용하십시오.

< TMS320LF240xA 타겟 보드와 RealDSP-UT 접속 방법 >



TMS320LF240xA는 BOOT_EN/XP(86) 신호가 Low이면 부트 모드로 동작하는데, IOPC2의 상태에 따라 SPI(High 상태) 또는 SCI(Low 상태) 부트로 동작한다. RealDSP-UT에서는 SCI 부트를 사용하므로 IOPC2 신호에 1K ~ 4.7K pull-down 처리를 해주어야 한다.

< TMS320F281x 타겟 보드와 RealDSP-UT 접속 방법 >



A) Boot선택 Switch 있는 보드인 경우

B) Boot선택 Switch가 없는 보드인 경우

■ TMS320F281x의 Boot Mode

GPIOF4 (SCITXDA)	GPIOF12 (MDXA)	GPIOF3 (SPISTEA)	SPIOF2 (SPICLK)	동작 모드
1	x	x	x	FLASH(0x3F7FF6) 실행
0	1	x	x	SPI 부트(Serial EEPROM) 실행
0	0	1	1	SCI 부트(SCI-A) 실행
0	0	1	0	H0 SRAM(0x3F8000) 실행
0	0	0	1	OTP(0x3D7800) 실행
0	0	0	0	GPIO B 포트 Parallel 부트 실행

TMS320F281x는 위의 표와 같이 리셋 동작 시 4개의 포트 상태에 따라 동작 모드를 결정한다. 본 회로에서는 FLASH ROM 실행 모드와 SCI 부트만을 사용하므로 3개의 핀(MDXA, SPISTEA, SPICLK)의 상태는 고정하고 1개의 핀(SCITXDA) 상태만 스위치로 선택하거나 또는 RealDSP-UT의 5번 Boot 핀을 사용하여 동작 모드를 결정하도록 함.

신호 처리 : MDXA = Low(pull-down)

SPISTEA = SPICLK = High(pull-up)

TXDA : 일부 로열시스의 CPU Module에서는 이 선택 핀이 없는 경우가 있습니다. 이는 개선된 모델로서 자동으로 처리하게 되었습니다.

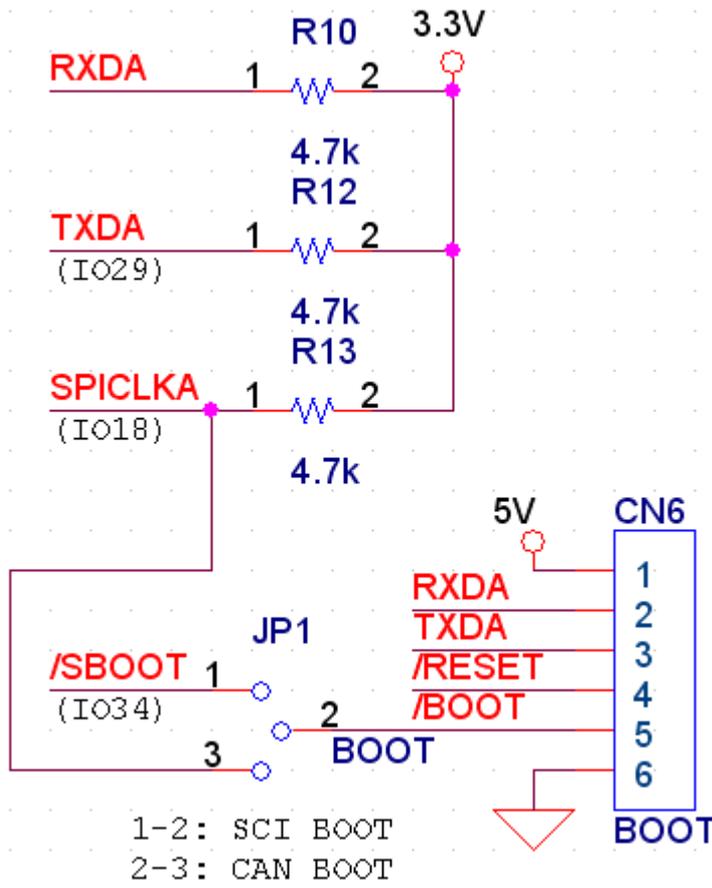
SW1 = 1-2(Low) FLASH ROM 실행

SW1 = 2-3(High) SCI 부트

주의 사항 :

1. 보드 외부에서 이 4개의 신호를 사용할 때 별도의 pull-up 또는 pull-down 처리에 유의한다. 만일 MDXA 핀에 별도의 pull-up 처리하거나 TXD-A를 pull-down 처리하면 원하는 동작 모드로 동작하지 않을 수 있다.
2. SCI 부트 또는 FLASH ROM을 라이트 할 때 반드시 SCI 부트 모드(SW1=2-3)로 한 상태에서 실행해야 되고, FLASH ROM 실행 시에 SW1=2-3으로 하고 리셋 동작을 실행한다.

< TMS320F280x 타겟 보드와 RealDSP-UT 접속 방법 >



■ TMS320F280x의 Boot Mode

GPIO18 SPICLKA SCITXB	GPIO29 SCITXA	GPIO34	동작 모드
1	1	1	FLASH(0x3F7FF6) 실행
1	1	0	SCI-A 부트 실행
1	0	1	SPI-A 부트 실행
1	0	0	I2C 부트 실행
0	1	1	eCAN-A mailbox 1 부트 실행
0	1	0	MO SRAM(0x000000) 실행
0	0	1	OTP FLASH(0x3D7800) 실행
0	0	0	Load Data from GPIO0~GPIO15

TMS320F280x는 위의 표와 같이 리셋 동작 시 3개의 포트 상태에 따라 동작 모드를 결정한다. 본 회로에서는 FLASH ROM 실행 모드와 SCI 부트만을 사용하므로 2개의 핀(SPICLKA, SCITXA)의 상태는 고정하고 1개의 핀(GPIO34) 상태만 스위치로 변경 사용하여 동작 모드를 결정하도록 함.

신호 처리 : SPICLKA = High(pull-up)

SCITXA = High(pull-up)

GPIO34 : Jumper Switch로 선택하도록 되어있음(기본은 FLASH Boot)

JP1 = 1-2 : SCI 부트 실행

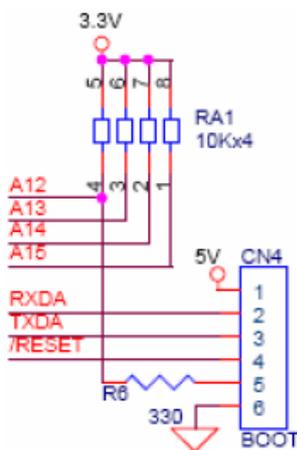
JP1 = 2-3 : CAN 부트 실행

주의 사항 :

1. 보드 외부에서 이 3개의 신호를 사용할 때 별도의 pull-up 또는 pull-down 처리에 유의한다.
2. SCI 부트 또는 FLASH ROM을 라이트 할 때 반드시 SCI 부트 모드로 한 상태에서 실행해야 되고, FLASH ROM 실행 시에 타겟을 FLASH 부트 모드로 둔 후 리셋한다.

< TMS320F28335 타겟 보드와 RealDSP-UT 접속 방법 >

MODE	GPIO87/XA15	GPIO86/XA14	GPIO85/XA13	GPIO84/XA12	MODE ⁽¹⁾
F	1	1	1	1	Jump to Flash
E	1	1	1	0	SCI-A boot
D	1	1	0	1	SPI-A boot
C	1	1	0	0	I2C-A boot
B	1	0	1	1	eCAN-A boot
A	1	0	1	0	McBSP-A boot
9	1	0	0	1	Jump to XINTF x16
8	1	0	0	0	Jump to XINTF x32
7	0	1	1	1	Jump to OTP
6	0	1	1	0	Parallel GPIO I/O boot
5	0	1	0	1	Parallel XINTF boot
4	0	1	0	0	Jump to SARAM
3	0	0	1	1	Branch to check boot mode
2	0	0	1	0	Branch to Flash, skip ADC calibration
1	0	0	0	1	Branch to SARAM, skip ADC calibration
0	0	0	0	0	Branch to SCI, skip ADC calibration



[RealDSP-UT or RealDSP-UT Academy POD 연결 회로]

5번핀(Boot)을 사용하여 위의 표에 있는 F, E 모드를 사용함

타겟 Reset시 5번핀이
High 일때 : Flash Boot Mode
Low 일때 : SCI-A Boot Mode

< Piccolo TMS320 F2802x/F2803x 타겟 보드와 RealDSP-UT 접속 방법 >

MODE	GPIO37/TDO	GPIO34/COMP2OUT/ COMP3OUT	$\overline{\text{TRST}}$	MODE
3	1	1	0	GetMode
2	1	0	0	Wait (see Section 3.3.10 for description)
1	0	1	0	SCI
0	0	0	0	Parallel IO
EMU	x	x	1	Emulation Boot

[Piccolo TMS320 F2802x/F2803x 부트 모드]

RealDSP-UT POD는 타겟 보드 연결 6핀 커넥터의 5번 /BOOT 신호를 가지고 평상시에는 타겟 보드를 내부 Flash 부트로 사용자 프로그램을 실행시키며 지정한 COFF 파일 사용자 프로그램을 다운로드 하는 경우에는 5번 /BOOT 핀 신호 레벨을 Low로 떨어뜨린 후 SCI 부트 동작을 수행합니다.

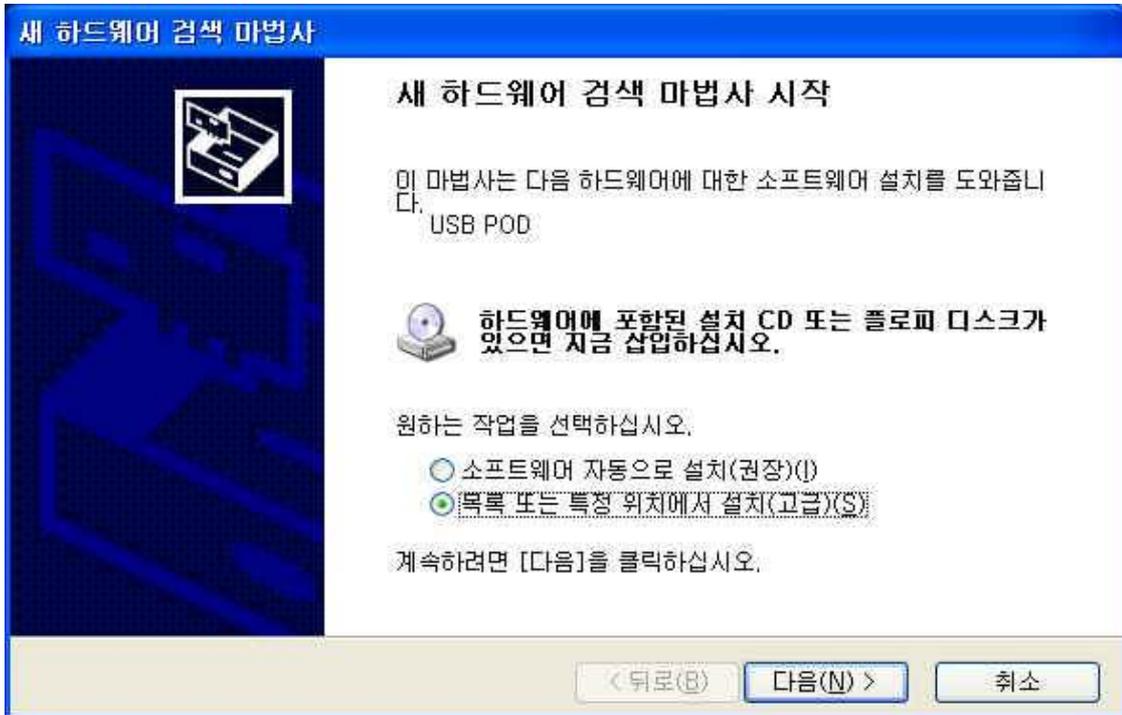
따라서 TI Piccolo TMS320 F2802x/F2803x 사용자께서는 GPIO37과 GPIO34 핀은 pull-up 처리하시고 /TRST 핀은 pull-down 하시길 바랍니다(즉, Flash Boot를 기본으로 설정). 그런 다음 아래 표와 같이 RealDSP-UT 연결 커넥터를 타겟 보드 MCU측에 연결하십시오.

RealDSP-UT 타겟 연결 6핀 커넥터	Piccolo TMS320 F2802x/F2803x MCU
1. VCC(DC 5V)	VCC(DC 5V)
2. TXD	GPIO28
3. RXD	GPIO29
4. /RESET	/RESET
5. /BOOT	GPIO37
6. GND	GND

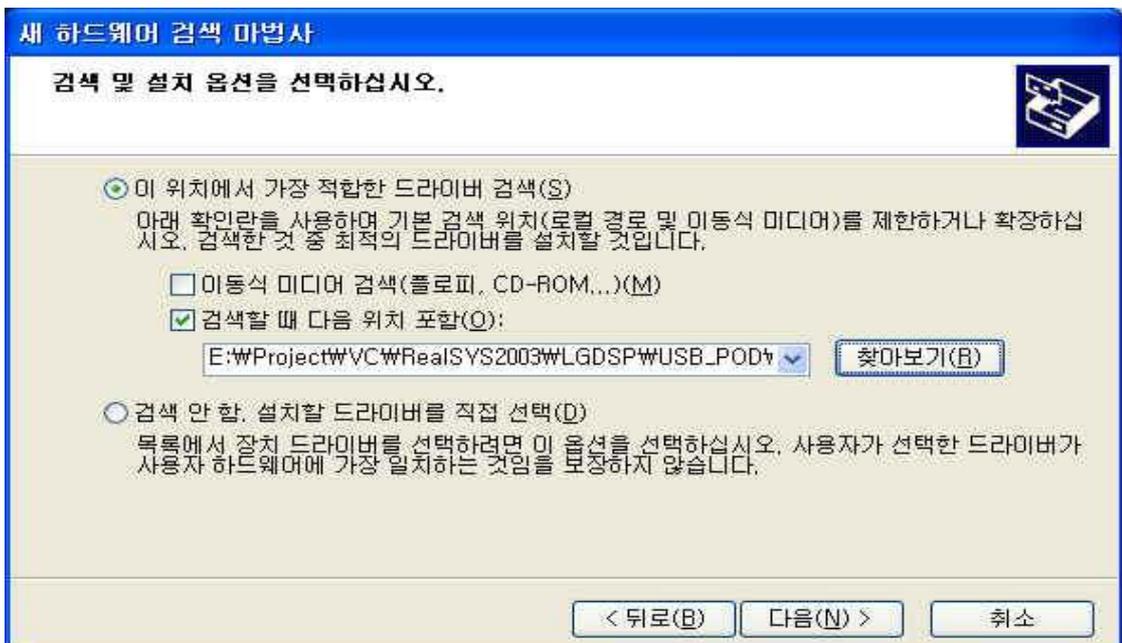
2. USB POD 디바이스 드라이버 설치

디바이스 드라이버를 설치하기에 앞서 먼저 구입시 제공된 CD에 USB POD용 Device Driver가 있는지 확인하시길 바랍니다.

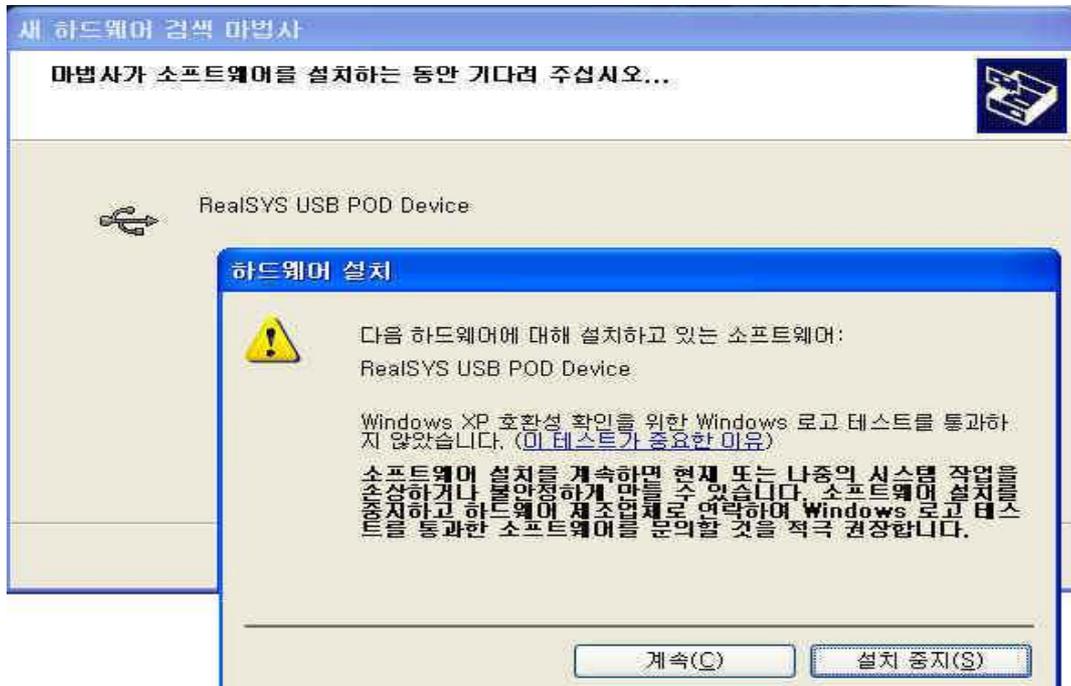
- 1) PC와 USB POD를 제공된 USB Cable로 연결하시면 다음과 같은 화면이 나옵니다.
이때 아래와 같이 “특정 위치에서 설치” 옵션을 선택하신 후 다음 버튼을 클릭하십시오.



- 2) 디바이스 드라이버가 있는 폴더 위치를 지정하신 후 다음 버튼을 클릭하십시오.



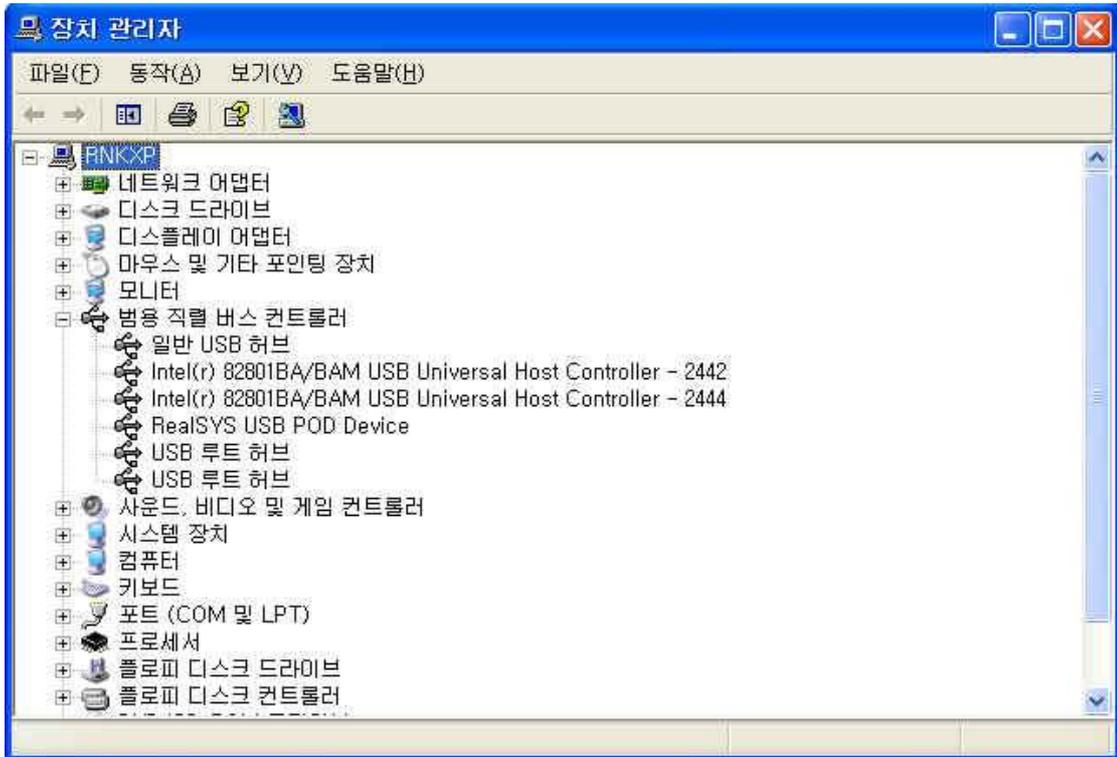
- 3) 제공된 디바이스 드라이버는 Windows의 Device Driver 호환 인증을 받지 않았습니다. 따라서 아래와 같은 화면이 나타나시면 “계속” 버튼을 클릭하신 후 설치를 계속하시길 바라며, 이는 제품 성능에 아무런 영향이 없으므로 사용자께서는 염려할 필요가 없습니다.



- 4) 위의 단계대로 하시면 일반적으로 디바이스 설치는 정상적으로 완료됩니다. 만약 설치 시 문제가 발생하면, 먼저 USB Cable이 정상적으로 연결되었는지, 또한 PC의 USB Host Controller가 정상적으로 동작하는지 확인해 주시길 바랍니다.



- 5) 정상적으로 디바이스 드라이버가 설치된 후 제어판의 장치 관리자를 보시면 아래의 그림과 같이 나타납니다.

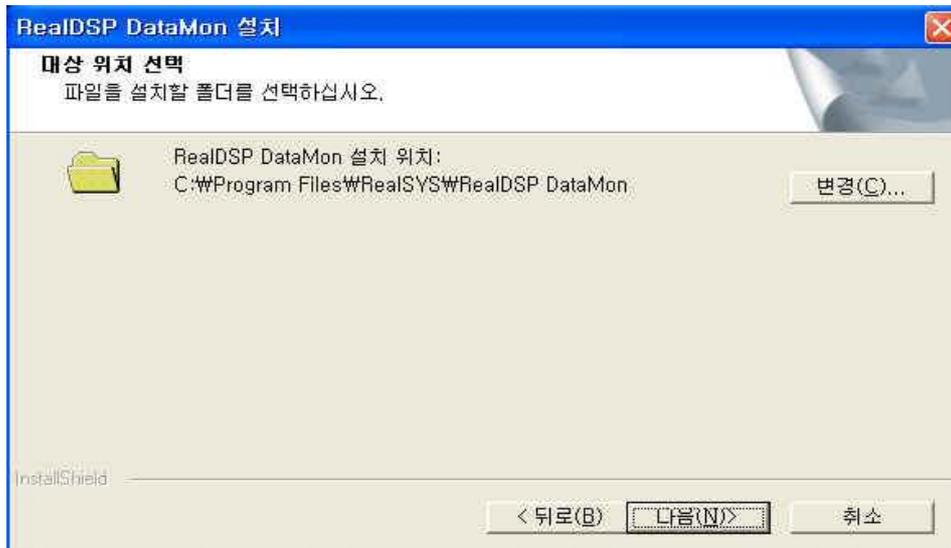


3. 운용 프로그램 설치

- 1) 제공된 CD에서 USB28X_v1.95.exe 아이콘을 더블 클릭하시면 설치가 시작됩니다.



- 2) 설치할 폴더를 지정하신 후 “다음” 버튼을 클릭하시면 정상적으로 설치가 완료되며, 바탕 화면에 단축 아이콘이 생깁니다.



4. RealDSP DataMon 프로그램 사용법

먼저 프로그램 사용하기에 앞서 위에서 설명한 하드웨어 설치를 참조하신 후 USB POD 및 사용자 타겟보드를 올바르게 연결하시길 바랍니다.

또한 사용자 프로그램을 다운로드하기 전에 다음 페이지에 있는 “환경 설정” 항목을 먼저 설정하시길 바라며, 현재 가지고 있는 RealDSP-UT POD의 Firmware 버전 및 RealDSP DataMon 프로그램의 버전을 확인하시길 바랍니다.

- RealDSP DataMon(Ver1.0 or Ver1.1) <-> RealDSP-UP POD의 Firmware(Ver1.0)
- RealDSP DataMon(Ver1.2) <-> RealDSP-UP POD의 Firmware(Ver1.1)
- RealDSP DataMon(Ver1.24) <-> RealDSP-UT POD의 Firmware(Ver1.2)
- RealDSP DataMon(Ver1.30) <-> RealDSP-UT POD의 Firmware(Ver1.2 or Ver1.3)
- RealDSP DataMon(Ver1.40) <-> RealDSP-UT POD의 Firmware(Ver1.2 or Ver1.3)
- **RealDSP DataMon(Ver1.50) <-> RealDSP-UT POD의 Firmware(Ver1.5)**
(TMS320F2801/2802/2806의 CSM Unlock 기능을 제외한 나머지 기능에 대해서는
POD Firmware Ver1.5 하위와 호환 가능)

< RealDSP DataMon 초기 화면 >

①

Symbol Name	Address	Data Type	Etc
eep_buf	0x0EA3	Array of Character	Array Info : [16;
EFP_ERR	0x0EA2	Unsigned charac...	
exit	0x1420	Integer	
freq	0x081F	Floating point	
i1_cnt	0x0800	Integer	
i2_cnt	0x0802	Integer	
i3_cnt	0x0803	Integer	
i4_cnt	0x0809	Integer	
i5_cnt	0x080A	Integer	
i6_cnt	0x080B	Integer	
k_vf	0x081D	Floating point	
key_code	0x080E	Unsigned charac...	
key_count	0x080F	Integer	
m_new	0x0804	Integer	
m_old	0x0806	Integer	
mcnt	0x081A	Integer	
nmi_cnt	0x080C	Integer	
prev_key_code	0x080D	Unsigned charac...	
sin_f	0x0A19	Array of Floating ...	Array Info : [500
sin_t	0x1980	Array of Integer	Array Info : [12,
speed	0x0805	Integer	
t4_cnt	0x0807	Unsigned integer	
table_dt	0x0821	Floating point	
table_u	0x0811	Integer	
table_v	0x0812	Integer	
table_w	0x0813	Integer	

②

Symbol Name	Value	View Type	Data Type
mcnt	10363	signed intege...	Integer
sin_f	0.000000	float (32bit)	Floating...
sin_t	328	signed intege...	Integer
speed	0	signed intege...	Integer
freq	45,043987	float (32bit)	Floating...

③

Address	Value	Ascii
0x0000	0x0000	..
0x0001	0x0001	..
0x0002	0x0002	..
0x0003	0x0003	..
0x0004	0x0007	..
0x0005	0x0000	..
0x0006	0x0000	..
0x0007	0x0007	..
0x0008	0x0008	..
0x0009	0x0009	..
0x000A	0x000A	..

④

Section Name	Start Address	Section Size
.text	0x0044	0x18A5 Words
.cinit	0x18E9	0x70 Words

⑤

준비 COM1 BPS : 115200 ON LINE Size : 0x3CDE TMS320LF2406A - 40MH;

① 환경 설정()

- 통신 방식 : 사용하고자 하는 PC와 USB POD사이의 통신 인터페이스를 설정
현재는 시리얼 및 USB를 지원합니다.
- 통신 포트 : 시리얼 인터페이스 선택 시 통신 포트
- 타임 아웃 : 데이터 모니터링 시 선택한 통신 인터페이스에서의 데이터 쓰기 및 읽기 타임 아웃 값
- 통신 속도 : USB POD와 DSP 타겟 보드 사이의 시리얼 디버깅 통신 BPS
< 데이터 모니터링을 위한 시리얼 디버깅 통신 BSP 설정에 관하여 >
 - TMS320LF240x/A
제공되는 debug_source 폴더의 debug_lf240x.c에서 #define BAUDRATE 정의에 선언된 통신 속도와 일치한 값으로 설정하시길 바랍니다.
 - TMS320F28x
제공되는 debug_source 폴더의 debug_28xxx.c에서 #define BAUDRATE 정의에 선언된 통신 속도와 일치한 값으로 설정하시길 바랍니다.
- Tx Delay : POD Firmware 버전 v1.2에서 새로 추가된 기능
데이터 모니터링 시 디버깅용으로 제공된 debug240x.c(or debug_serial.c)가 사용자 프로그램에 따라서 interrupt 응답 시간이 늦는 경우가 발생하여 제대로 데이터 모니터링이 안되는 경우에 Tx Delay 값을 알맞게 설정하십시오.
또한 위와 같은 현상이 발생하면 통신 속도를 낮추어서 사용하시길 바랍니다.
- 시간 간격 : 데이터 모니터링 시 데이터 수집 간격 설정
- 타겟 DSP : 사용자 타겟 보드의 CPU 타입 설정

- 현재 POD의 펌웨어 확인 : 현재 연결된 RealDSP-UT POD의 Firmware Version을 체크함.

< CSM & Erase Options for TMS320F28x/LF240xA Device >

- Erase Option (TMS320F28x Device에서만 지원)

TMS320F28x의 내부 플래시 롬은 Sector로 구성되어있으며, 따라서 Erase 시 Sector 별로 지울 수 있습니다. 또한 사용자께서는 내부 플래시 롬의 일부 영역을 고정된 Data 영역으로 사용하여 자주 지울 필요가 없는 경우에는 사용자 프로그램 Write 시 이 영역을 지울 필요가 없으므로 다음과 같이 두 가지 옵션을 두었습니다.

➤ Intelligent Erase :

읽어 들인 COFF 파일의 섹션 정보를 바탕으로 전체 플래시 롬 Sector 영역 중 실지로 Write 하는 Sector 영역만 지웁니다.

➤ Erase All :

무조건 전체 플래시 롬 Sector 영역을 지웁니다.(최대 25sec 걸림)

- Unlock Key Code Option

TMS320LF240xA/TMS320F28x는 CSM이라는 기능이 있는데 이 기능은 Secure Memory 영역에 대하여 Code Security를 사용자가 설정할 수 있으며, 만약 CSM이 설정된(Secured Device라고 부름) 타겟 보드에 대해서는 이 영역의 메모리에 대해 외부로 부터 Read Access 할 수 없습니다. 일반적으로 개발 시에는 사용하지 않으며, 제품 기능이 완성되고 제품 양산 시 지적 재산권 보호 차원에서 이 기능을 사용합니다.

만약 타겟 보드의 CPU가 Secured된 Device이며 다시 사용자 프로그램을 내부 Flash ROM에 Write하고자 하는 경우에는 이 옵션을 사용하며, 또한 아래에 있는 4개(TMS320LF240xA) 또는 8개(TMS320F28x)의 KEY 값(Password)을 올바르게 입력하시길 바랍니다. 입력한 KEY 값은 Flash Algorithm의 CSM Unlock에 사용됩니다.

< 주의 >

만약 TMS320F28x에서 8개의 Password들 모두를 0x0000으로 할 경우에는 그 Device는 Permanent Secured가 되어 다시 프로그램을 Update할 수 없습니다.

< 기능 개선 >

TMS320F2801/2802/2806 MCU에 대해서 Ver1.4에서는 CSM Unlock 기능이 지원되지 않는 점이 있었는데 이번 버전에서 이 문제를 해결함. 이들 MCU를 사용하는 사용자께서는 현재 사용하고 있는 RealDSP-UT 28x POD의 Firmware 버전을 체크하여 Ver1.5 이상인지 확인 후 사용하시길 바랍니다.

- TMS320LF240xA Device를 CSM Locked(Secured) 하는 방법
될 수 있는 한 제품 개발 도중에는 사용하지 마시고, 제품 양산 시 사용하시길 바랍니다.

```
- csmpwd.asm -----
.sect "csmpwd"
.word 0000h      ; PWL0
.word 0000h      : PWL1
.word 0000h      ; PWL2
.word 0000h      ; PWL3
.end
-----
```

위 Assembly Code를 사용자 프로그램에 추가한다.

```
- dsp2406A.cmd -----
MEMORY
{
    PAGE 0: /* Program Memory */
        ...
        CSM_PWD      : org = 0040h, len = 0004h
        ...
}

SECTIONS
{
    ...
    csmpwd: > CSM_PWD    PAGE = 0 /* CSM Password */
    ...
}
-----
```

[알림]

현재 TMS320LF240xA는 TMS320F28x와 같이 Device가 Secured된 상태인지를 나타내는 플래그가 없어 Flash Algorithm에서 CSM Unlock Match Flow 과정에서 Unlocking을 하여도 디바이스가 Unlock되었는지 알 수 없어 계속 Erase & Write 과정을 하게 되어 사용자 입장에서는 마치 디바이스가 제대로 Locking되지 않았다고 생각될 수 있습니다. 테스트 결과 RealDSP-UT에서 Secured된 디바이스에 일치하지 않은 Password를 입력하여 라이트하는 경우에는 RealDSP DataMon 프로그램에서 Erase & Write가 되어도 실제로 프로그램이 변경되지 않으며 일치된 Password를 입력할 시에는 프로그램이 변경되는 것을 확인할 수 있습니다. 이를 통해 사용자께서는 라이트가 된 Device가 Lock이 걸렸는지 확인할 수 있습니다.

- TMS320F28x Device를 CSM Locked(Secured) 하는 방법
될 수 있는 한 제품 개발 도중에는 사용하지 마시고, 제품 양산 시 사용하시길 바랍니다.
< 제공되는 예제의 DSP281x_CSMPasswords.asm과 DSP2812.CMD을 참조 바람 >

```

- DSP281x_CSMPasswords.asm -----
.sect "csmpasswd"
.int 0xFFFF      ; PWL0
.int 0xFFFF      : PWL1
.int 0xFFFF      ; PWL2
.int 0xFFFF      ; PWL3
.int 0xFFFF      ; PWL4
.int 0xFFFF      ; PWL5
.int 0xFFFF      ; PWL6
.int 0xFFFF      ; PWL7

.sect "csm_rsvd"
.loop (3F7FF5h - 3F7F80h + 1)
    .int 0x0000
.endloop
-----

```

위 Assembly Code를 사용자 프로그램에 추가한다.

```

- DSP2812.CMD -----
MEMORY
{
    ...
    CSM_RSVD      : origin = 0x3F7F80, length = 0x000076
    CSM_PWL       : origin = 0x3F7FF8, length = 0x000008
    ...
}

SECTIONS
{
    ...
    csm_rsvd      : > CSM_RSVD, PAGE = 0
    csmpasswd     : > CSM_PWL,  PAGE = 0
    ...
}
-----

```

② Toolbar : RealDSP DataMon 프로그램의 주요 기능에 대한 단축 아이콘



- COFF 파일 열기()

타겟 보드에 다운로드될 사용자 프로그램 실행 파일(*.out)인 COFF 파일을 읽을 때 사용하며, COFF 파일을 읽고 난 후 ②창에는 COFF파일에 정의된 심볼 정보를 표시합니다. 또한 ⑤번의 섹션 정보에는 COFF 파일의 각각의 섹션 항목 및 데이터 정보가 표시됩니다.
- COFF 파일 Reload()

최근에 열었던 COFF 파일을 다시 엽니다.
- 심볼 파일 열기()

그전에 사용자가 모니터링을 하기 위해 ③창에 등록된 변수 심볼 정보를 저장한 파일 (*.sym)을 읽을 때 사용합니다.
- 심볼 파일 저장()

③창에 등록된 모니터링 변수 심볼 정보를 파일로 저장합니다.
- SCI 부트 ()

 - Target CPU가 TMS320LF240xA인 경우

현재 읽어 들인 사용자 프로그램(COFF 파일의 프로그램 데이터)을 SCI Boot 모드 방식으로 타겟 보드의 내부 Flash Memory 영역에 Write한 후, 타겟보드를 리셋합니다.
 - Target CPU가 TMS320F28x인 경우

현재 읽어 들인 사용자 프로그램이 RAM 영역에 다운로드할 내용만 있다면 SCI Boot 모드 방식으로 타겟 보드의 내부 RAM영역에 Write한 후 TMS320F281x의 경우 H0 그리고 TMS320F280x의 경우 M0 스타트 어드레스로 부터 프로그램을 실행시킵니다. 또한 읽어 들인 사용자 프로그램이 Flash or OTP ROM 영역에 다운로드할 내용이 있으면 SCI Boot 모드 방식으로 먼저 Flash Algorithm Code를 RAM영역에 다운로드한 후, 사용자 프로그램을 Flash or OTP ROM 영역에 write 합니다.

Flash 및 OTP ROM에 write가 제대로 되었는지 확인하고자 하면, Target Board를 Flash 또는 OTP Boot Mode로 설정하신 후, Target Board를 리셋하시길 바랍니다.

사용자께서는 이 기능을 사용하기에 앞서 먼저 “환경 설정”을 통하여 사용하고자 하는 환경을 설정하시길 바랍니다.

- 변수 모니터링()
 - ③창에 등록된 변수 심볼에 대하여 주기적으로 모니터링 시작 또는 중지합니다.
- 주소 모니터링()
 - ④창에 연속적인 주소에 대하여 주기적으로 모니터링 시작 또는 중지합니다.
- 타겟 리셋()
 - 타겟보드를 리셋합니다.
- 통신포트 닫기()
 - 현재 연결된 PC측 통신 포트(Serial 또는 USB)를 닫습니다.
- 메모리 그래프()
 - 사용자 프로그램이 실행되는 동안 메모리의 일정 영역을 모니터링하여 값을 얻은 후 이들 값을 그래프로 표시해줍니다. 주로 데이터 테이블이나 Debugging을 위해 선언해 둔 일정 메모리 배열 영역을 최대 선택한 4개 심볼 항목에 대해 그래프로 표시하여 사용합니다.

그래프 환경 설정 ✕

Lable	Address	Length	Type	Precision
cv	0x08ED	0x64	float	3
sv	0x0825	0x64	float	3

Ch1 Line Color Ch2 Line Color

Ch3 Line Color Ch4 Line Color

개별적 그래프 그리기 추가 대체 삭제

심볼 리스트 sv 레이블 이름 sv

시작 주소 0x0825 데이터 길이 100

데이터 타입 float Precision 3

Scale Mode 자동 Scale Min -10 Max 10

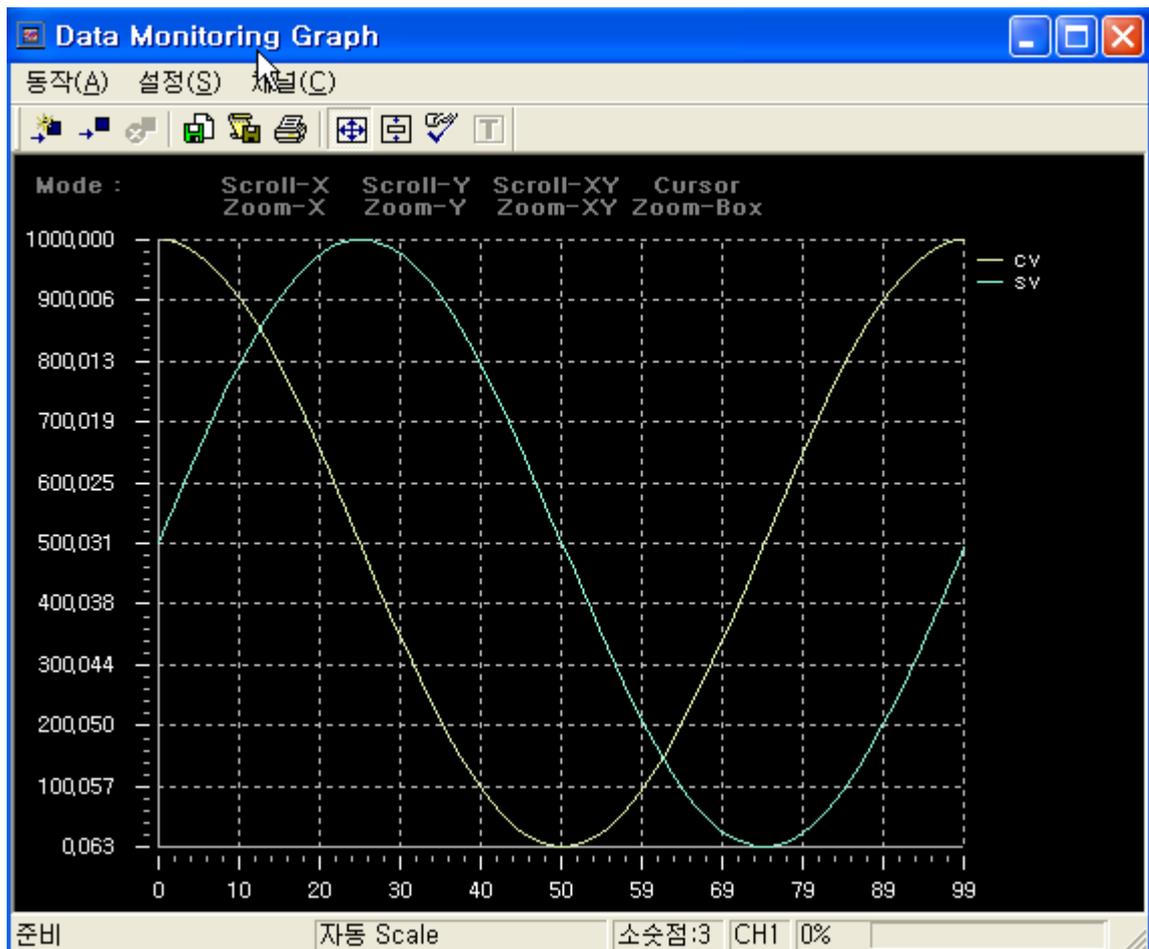
확인 취소

- 심볼 리스트 : 현재 읽어 들인 COFF 파일에서 얻은 변수 심볼 목록
- 레이블 이름 : 그래프 창에서 입력한 데이터를 대표할 이름
- 시작 주소 : 모니터링 하고자 하는 메모리 시작 주소

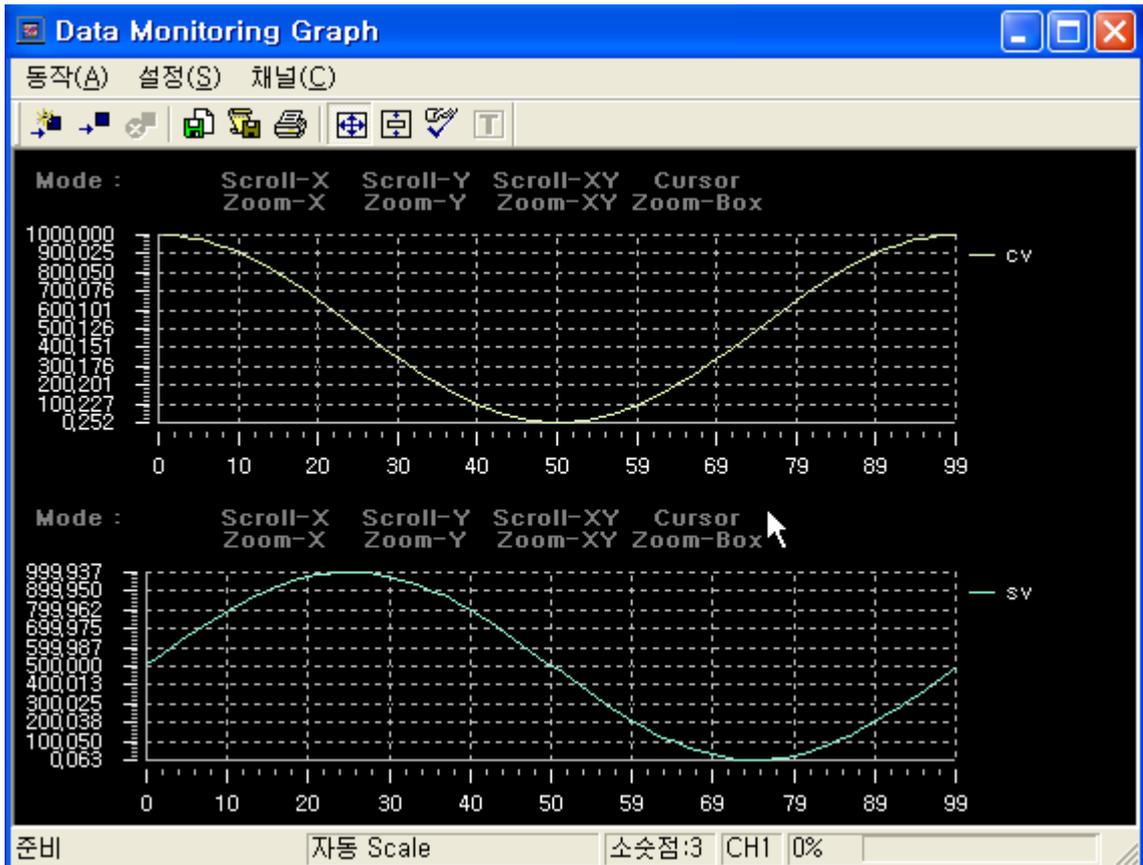
- 데이터 길이 : 시작 주소에서 메모리의 어느 영역만큼 데이터를 수집할 지 결정
- 데이터 타입 : 그래프에 표시될 데이터 값 종류
- Precision : 위의 데이터 타입이 Float인 경우 소수점 몇까지 표시할 것인가 결정
- Scale Mode : 그래프 창의 Y축 값을 자동으로 Scale할 것인가? 또는 사용자가 정의한 최대, 최소값을 기준으로 할 것인가?
- Min : 위의 Scale Mode가 수동 모드일 때 Y축 최대값
- Max : 위의 Scale Mode가 수동 모드일 때 Y축 최소값
- Channel Line Color : 최대 4개까지 위에서부터 순서대로 선택한 심볼에 대한 그래프 라인의 색상을 지정하는데 사용.
- 개별적 그래프 그리기 : 최대 4개까지 위에서부터 순서대로 선택한 심볼에 대해서 하나의 그래프에 표시할 지 아니면 개별 그래프에 표시할지 설정.

사용자가 입력한 그래프 설정 데이터를 “추가”, “대체” 그리고 “삭제” 버튼을 이용하여 리스트에 등록하십시오. 그런 후 등록된 리스트 항목 중 그래프로 표시할 항목을 선택하고 나서 확인 버튼을 누르면 데이터 수집 및 그래프 창이 나타납니다. 현재는 최대 4개까지의 항목을 동시에 그래프로 표시할 수 있습니다.

< 개별적 그리기 모드 : Unchecked >



< 개별적 그리고 모드 : Checked >



시간 그래프 ()

선택한 변수 심볼 및 주소에 대하여 일정 시간 간격으로 데이터를 계속 수집하며, 이를 최대 선택한 4개 심볼 항목에 대해서 그래프로 표시합니다.

The screenshot shows a dialog box titled "시간 그래프 환경 설정" (Time Graph Environment Settings). It contains a table with the following data:

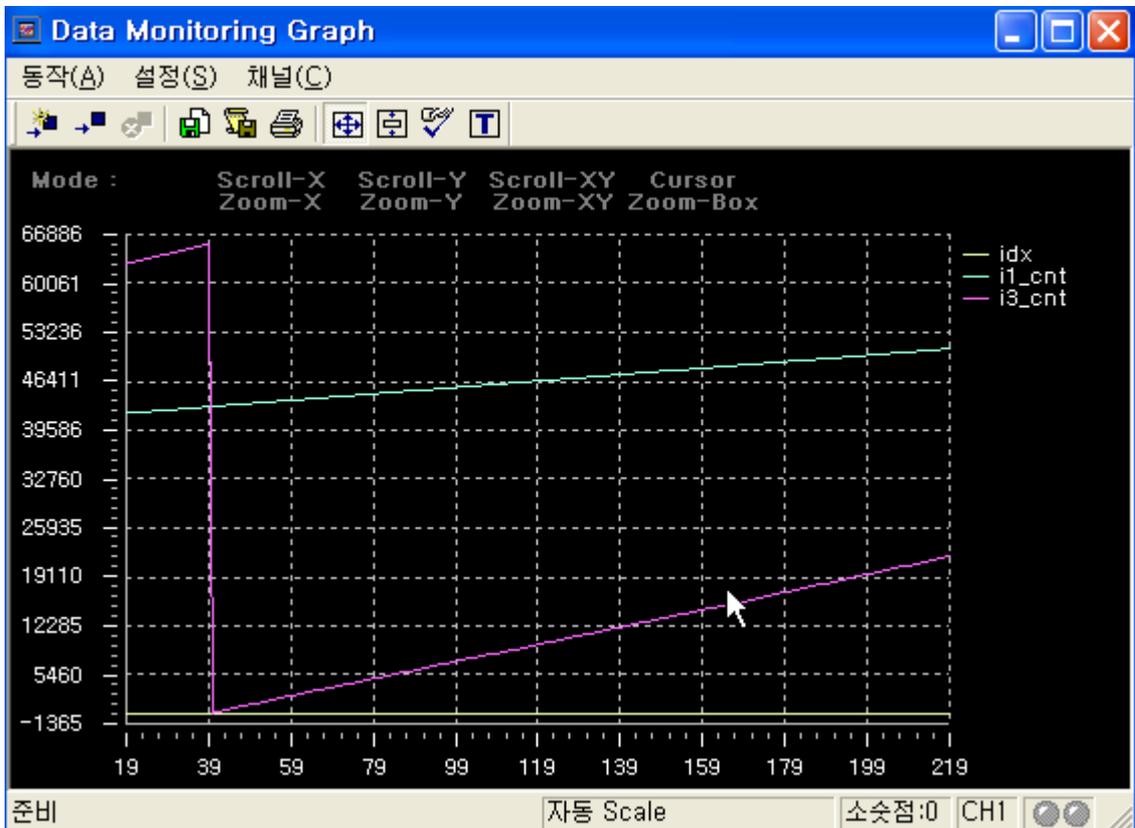
Lable	Address	Type	Precision
idx	0x0805	integer	0
i1_cnt	0x0800	unsigned integer	0
i3_cnt	0x0803	unsigned integer	0

Below the table, there are color selection options for Ch1 (yellow), Ch2 (cyan), Ch3 (magenta), and Ch4 (grey). There are also buttons for "추가" (Add), "대체" (Replace), and "삭제" (Delete). A checkbox labeled "개별적 그래프 그리기" (Draw individual graphs) is present. Below this, there are input fields for "심볼 리스트" (Symbol List) set to "mcnt", "레이블 이름" (Label Name) set to "mcnt", "시작 주소" (Start Address) set to "0x0813", and "데이터 타입" (Data Type) set to "unsigned integer". At the bottom, there are "Scale Mode" (set to "자동 Scale"), "Min" (-10), "Max" (10), and "시간 간격" (Time Interval) set to "10 (ms)". Buttons for "확인" (OK) and "취소" (Cancel) are at the bottom right.

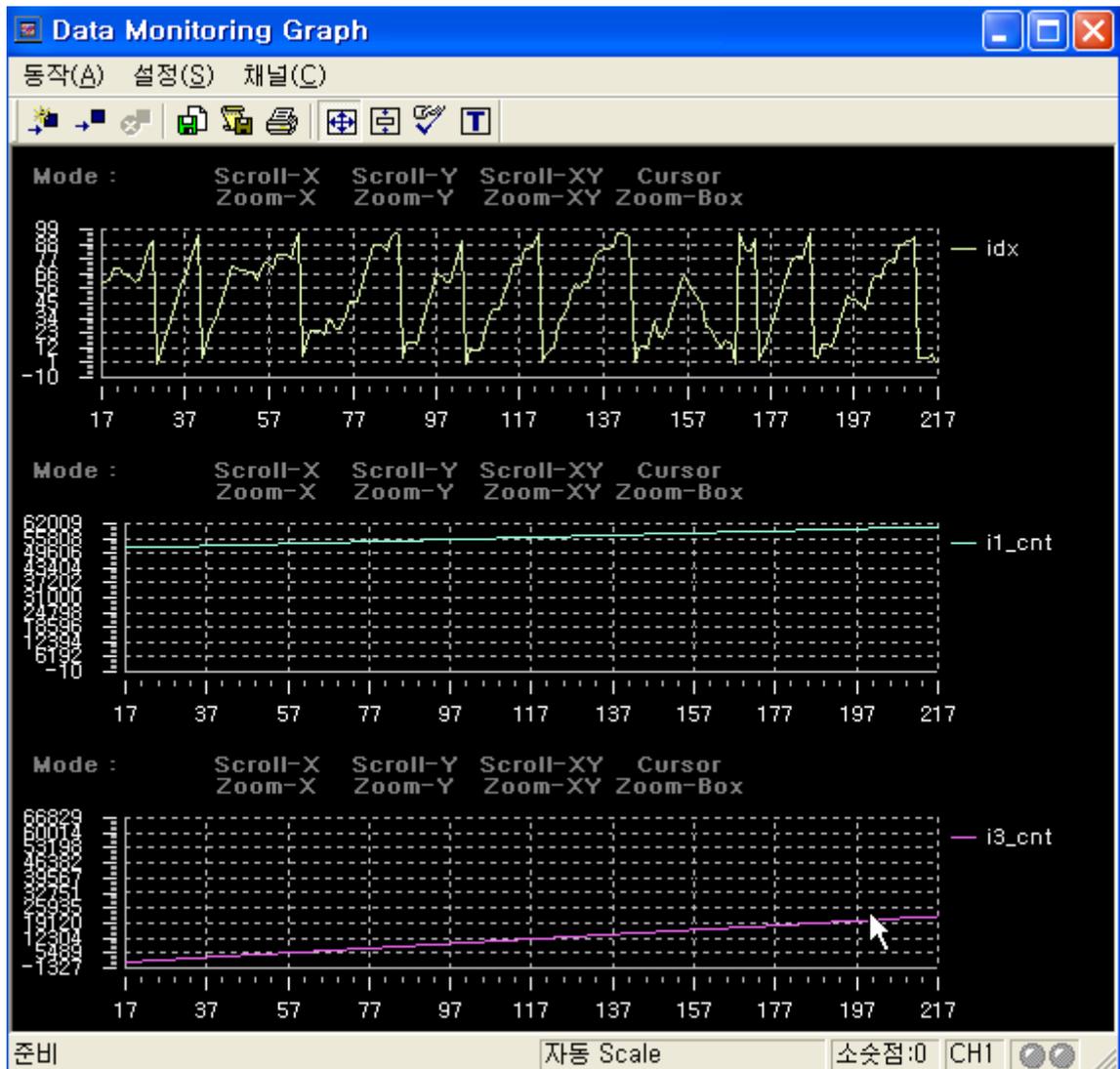
- 심볼 리스트 : 현재 읽어 들인 COFF 파일에서 얻은 변수 심볼 목록
- 레이블 이름 : 그래프 창에서 입력한 데이터를 대표할 이름
- 시작 주소 : 모니터링 하고자 하는 메모리 시작 주소
- 데이터 타입 : 그래프에 표시될 데이터 값 종류
- Precision : 위의 데이터 타입이 Float인 경우 소수점 몇까지 표시할 것인가 결정
- Scale Mode : 그래프 창의 Y축 값을 자동으로 Scale할 것인가? 또는 사용자가 정의한 최대, 최소값을 기준으로 할 것인가?
- Min : 위의 Scale Mode가 수동 모드일 때 Y축 최대값
- Max : 위의 Scale Mode가 수동 모드일 때 Y축 최소값
- 시간 간격 : 데이터 수집 시간 간격을 설정
- Channel Line Color : 최대 4개까지 위에서부터 순서대로 선택한 심볼에 대한 그래프 라인의 색상을 지정하는데 사용.
- 개별적 그래프 그리기 : 최대 4개까지 위에서부터 순서대로 선택한 심볼에 대해서 하나의 그래프에 표시할 지 아니면 개별 그래프에 표시할지 설정

사용자가 입력한 그래프 설정 데이터를 “추가”, “대체” 그리고 “삭제” 버튼을 이용하여 리스트에 등록하십시오. 그런 후 등록된 리스트 항목 중 그래프로 표시할 항목을 선택하고 나서 확인 버튼을 누르면 데이터 수집 및 그래프 창이 나타납니다. 현재는 최대 4개까지의 항목을 동시에 그래프로 표시할 수 있습니다.

< 개별적 그리기 모드 : Unchecked >



< 개별적 그리기 모드 : Checked >



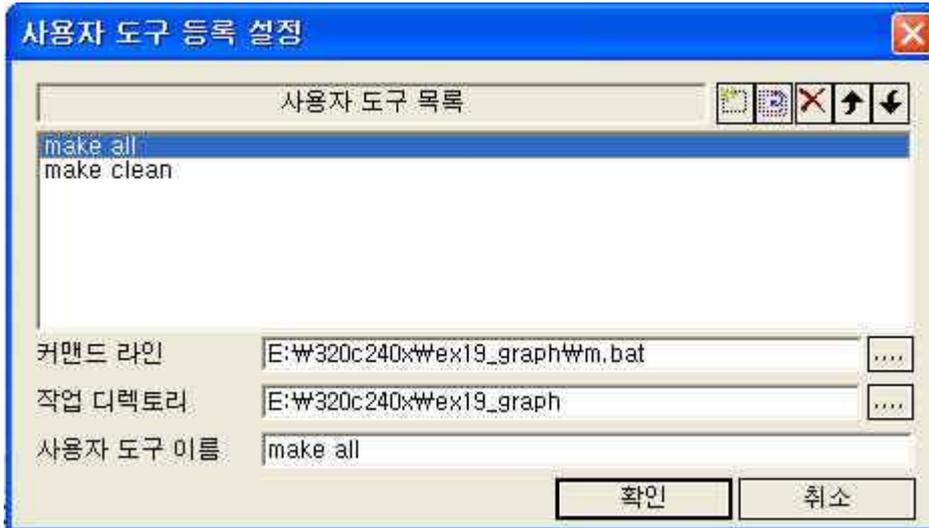
➤ 환경 설정()

RealDSP DataMon 프로그램의 주요 환경 변수를 설정합니다.

따라서 사용자께서는 RealDSP DataMon 프로그램의 주요 기능을 사용하기에 앞서 반드시 “환경 설정”을 해주시길 바랍니다.

➤ 사용자 도구 등록()

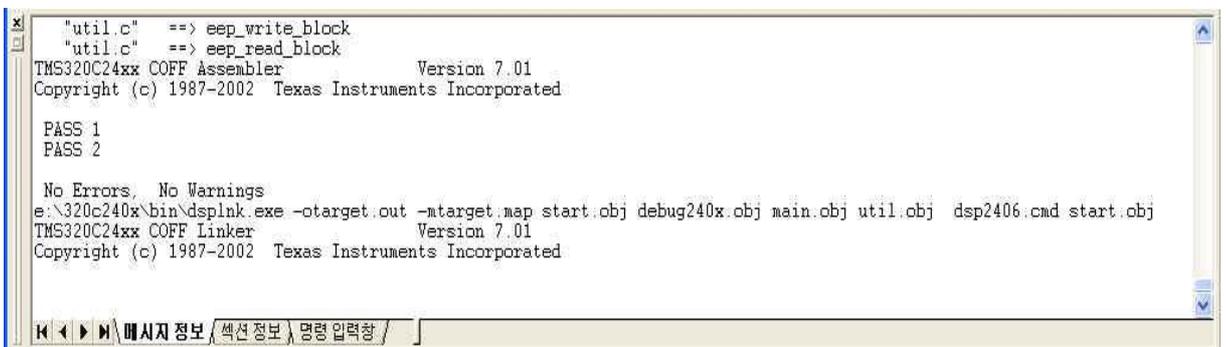
사용자가 자주 쓰는 외부 프로그램(예, 계산기 또는 compiling & linking 관련 batch 파일 등...)을 최대 5개까지 등록하여 사용할 수 있으며, 이는 Ultra-Editor의 Tool Configuration 기능과 유사합니다.



- 커맨드 라인 : 등록할 프로그램의 실행 파일 경로 및 인수 입력 창으로서 사용자께서는 옆에 있는 버튼(...)을 클릭하여 쉽게 등록할 실행 파일 경로를 찾아 입력하신 후 실행 인수를 입력하시면 됩니다.
- 작업 디렉토리 : 선택한 프로그램 실행 시 작업 대상 디렉토리 입력 창
- 사용자 도구 이름 : 선택한 프로그램을 대표할 단축 아이콘의 이름 입력 창

- 
 - 추가 : 사용자가 입력한 정보를 리스트에 추가할 때 사용
 - 대체 : 그전에 입력한 정보를 수정한 내용으로 대체할 때 사용
 - 삭제 : 그전에 입력한 정보를 리스트에서 삭제할 때 사용
 - 위로 : 리스트에서 현재 선택한 Tool의 순위를 아래로 이동시킬 때 사용
 - 아래로 : 리스트에서 현재 선택한 Tool의 순위를 위로 이동시킬 때 사용

등록된 사용자 Tool들에 대해서는 단축 아이콘()이 생성되며, 만약 등록된 프로그램이 DOS 기반 프로그램인 경우에는 표준 Input/Output 메시지를 Capture하여 ⑤의 메시지 정보 창에 표시합니다.



➤ 사용자 정의 툴바()

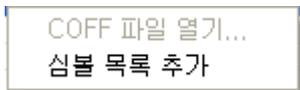
위의 사용자 도구 등록()에서 등록한 Tool들을 대표하는 단축 아이콘 모음들

③ 심볼 정보 창 : COFF파일에 정의된 사용자 프로그램 관련 변수 심볼 정보 표시

Symbol Name	Address	Data Type	Etc
ECanaMOTORegs	0x0060C0	Structure	
ECanaMOTSTRegs	0x006080	Structure	
ECanaRegs	0x006000	Structure	
enc_10ms	0x00814A	Integer	
EvaRegs	0x007400	Structure	
EvbRegs	0x007500	Structure	
exit	0x3F8B77	Integer	
FlashRegs	0x000A80	Structure	
FP	0x000002	Integer	
GpioDataRegs	0x0070E0	Structure	
GpioMuxRegs	0x0070C0	Structure	
key	0x00814B	Integer	
key_code	0x008154	Unsigned character	
key_count	0x008140	Integer	
m_new	0x008144	Integer	
m_old	0x008145	Integer	
McbspaRegs	0x007800	Structure	
mcnt	0x008146	Integer	
msg	0x3F8E4C	Array of Character	Array Info : [16]
PieCtrlRegs	0x000CE0	Structure	
PieVectTable	0x000D00	Structure	
PieVectTableInit	0x3F8D4C	Structure	
prev_key_code	0x008149	Unsigned character	
rpm	0x008142	Integer	
SciaRegs	0x007050	Structure	
ScjbRegs	0x007750	Structure	
SpiaRegs	0x007040	Structure	

● 변수 심볼 선택 후 Double-Click 시
 선택한 변수 심볼을 ③의 심볼 모니터링 창에 등록 후 모니터링을 시작합니다.

● 왼쪽 마우스 버튼 Click 시
 아래와 같은 Pop-up 메뉴가 나타납니다.



- COFF 파일 열기... : 새로운 COFF파일을 열 때 사용
- 심볼 목록 추가 : 선택한 변수 심볼들을 ③의 심볼 모니터 창에 등록할 때 사용

④ 심볼 모니터링 창 : 사용자가 선택한 변수 심볼에 대한 모니터링 결과 표시 창

Symbol Name	Value	View Type	Data Type
mcnt	0x465B	hex decima	
sin_f	0.000000	float (32bit)	
sin_t	1061	signed integ	
speed	0	signed integ	
freq	44,985336	float (32bit)	

심볼 파일 열기...

심볼 파일 저장...

변수 모니터링 중지

선택 항목 삭제

전체 항목 삭제

데이터 변경...

데이터 보기 변환 ▶

Hex Decimal (16bit)

Hex Decimal (32bit)

Signed Integer (16bit)

Signed Integer (32bit)

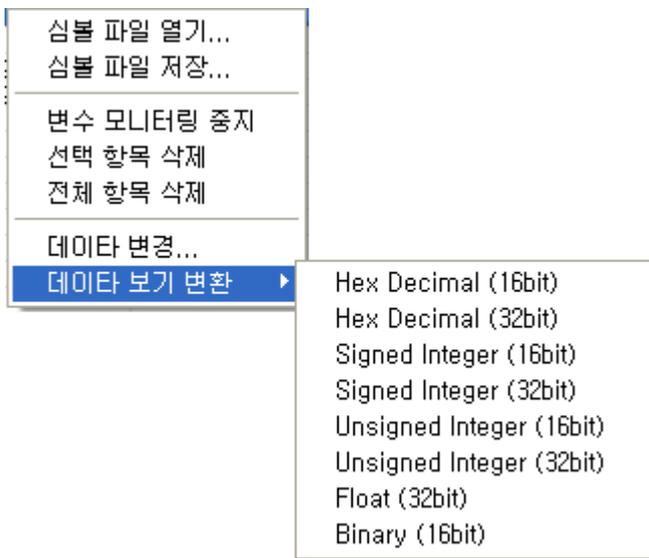
Unsigned Integer (16bit)

Unsigned Integer (32bit)

Float (32bit)

Binary (16bit)

- 모니터링 변수 심볼 선택 후 Double-Click 시
선택한 변수의 데이터 값을 사용자가 입력한 값으로 변경할 때 사용
- 왼쪽 마우스 버튼을 Click 시
아래와 같은 Pop-up 메뉴가 나타납니다.



- 심볼 파일 열기... : 이전에 저장된 변수 심볼 리스트 정보 파일을 엽니다.
- 심볼 파일 저장... : 현재 등록된 변수 심볼 리스트 정보를 파일에 저장합니다.
- 변수 모니터링 시작/중지 : 등록된 변수 심볼에 대한 모니터링을 시작/중지 합니다.
- 선택 항목 삭제 : 선택한 변수 심볼을 삭제합니다.
- 전체 항목 삭제 : 리스트에 있는 변수 심볼을 삭제합니다.
- 데이터 변경... : 선택한 변수의 데이터를 사용자가 입력한 값으로 변경합니다.
- Hex Decimal : 모니터링 결과를 Hex Decimal로 표시합니다.

- Signed Integer : 모니터링 결과를 Signed Integer로 표시합니다.
- Unsigned Integer : 모니터링 결과를 Unsigned Integer로 표시합니다.
- Float : 모니터링 결과를 Float로 표시합니다.
- Binary : 모니터링 결과를 Binary로 표시합니다.

⑤ 주소 모니터링 창 : 사용자가 선택한 주소에 대한 모니터링 결과 표시 창

Address	Value	Ascii	
0x0000	0x0000	..	
0x0001	0x0001	..	주소 모니터링 중지 시작 주소 변경...
0x0002	0x0002	..	
0x0003	0x0003	..	
0x0004	0x0007	..	데이터 변경...
0x0005	0x0000	..	데이터 보기 변환 ▶
0x0006	0x0000	..	Hex Decimal (16bit) Hex Decimal (32bit) Signed Integer (16bit) Signed Integer (32bit) UnSigned Integer (16bit) Unsigned Integer (32bit) Float (32bit) Binary (16bit)
0x0007	0x0007	..	
0x0008	0x0008	..	
0x0009	0x0009	..	
0x000A	0x000A	..	
0x000B	0x000B	..	
0x000C	0x000C	..	
0x000D	0x000D	..	
0x000E	0x000E	..	
0x000F	0x000F	..	
0x0010	0x0010	..	
0x0011	0x0011	..	

- 모니터링 주소 선택 후 Double-Click 시
선택한 주소의 데이터 값을 사용자가 입력한 값으로 변경할 때 사용
- 왼쪽 마우스 버튼을 Click 시
아래와 같은 Pop-up 메뉴가 나타납니다.



- 주소 모니터링 시작/중지 : 등록된 주소에 대한 모니터링을 시작/중지 합니다.
- 시작 주소 변경...: 모니터링 시작 주소를 입력합니다.
- 데이터 변경... : 선택한 주소의 데이터를 사용자가 입력한 값으로 변경합니다.
- Hex Decimal : 모니터링 결과를 Hex Decimal로 표시합니다.
- Signed Integer : 모니터링 결과를 Signed Integer로 표시합니다.
- Unsigned Integer : 모니터링 결과를 Unsigned Integer로 표시합니다.
- Float : 모니터링 결과를 Float로 표시합니다.
- Binary : 모니터링 결과를 Binary로 표시합니다.

⑥ 메시지 창

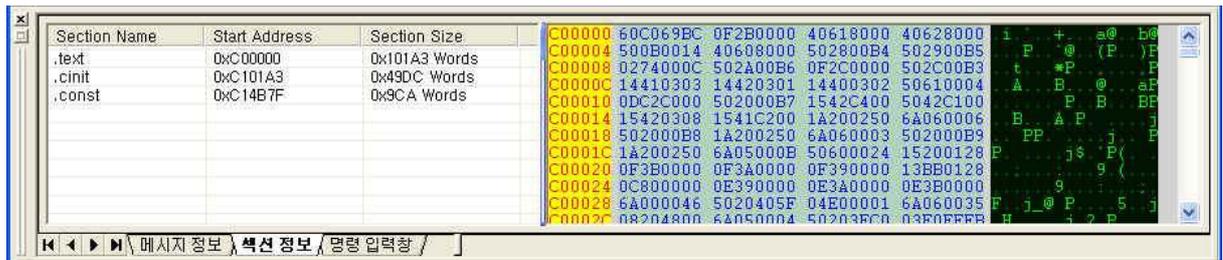
- 메시지 정보

등록한 사용자 Tool()이 DOS 기반 프로그램인 경우에는 표준 Input/Output 메시지를 Capture하여 표시하는 창입니다.



- 섹션 정보

사용자 선택한 COFF파일에 있는 각각의 섹션 정보 및 그 섹션에 해당하는 프로그램 데이터를 표시하는 창입니다.



왼쪽에 있는 섹션 정보 리스트의 각 항목을 Double-Click하면 선택한 섹션의 프로그램 데이터가 왼쪽 창에 표시됩니다.