

RealView[®] Development Suite

버전 3.0

시작 설명서

ARM[®]

RealView Development Suite

시작 설명서

Copyright © 2003-2006 ARM Limited. All rights reserved.

릴리스 정보

이 설명서에서 변경된 내용은 다음과 같습니다.

변경 내역

날짜	발행판	변경 내용
2003년 9월	A	RVDS 버전 2.0 릴리스
2004년 1월	B	RVDS 버전 2.1 릴리스
2004년 12월	C	RVDS 버전 2.2 릴리스
2005년 5월	D	RVDS 버전 2.2 SP1 릴리스
2006년 3월	E	RVDS 버전 3.0 릴리스

소유권 고지 사항

® 또는 ™ 표시가 있는 단어와 로고는 ARM Limited 소유의 등록 상표 또는 상표입니다. 이 설명서에 언급된 기타 브랜드와 이름은 해당 소유자의 상표일 수 있습니다.

이 설명서에 포함된 전체 또는 일부 정보나 설명된 제품은 해당 저작권 소유자의 사전 서면 승인 없이는 어떤 형태로도 개조되거나 복제될 수 없습니다.

이 설명서에 설명된 제품은 지속적으로 개발 및 개선될 수 있습니다. 이 설명서에 포함된 모든 제품 명세와 해당 사용법은 ARM의 신뢰하에 제공됩니다. 그러나 ARM에서는 상품성 또는 특정 목적에의 적합성을 비롯하여 그 밖의 목적이거나 명시적인 모든 보증을 부인합니다.

이 설명서는 제품 사용자를 지원하는 용도로만 만들어졌습니다. ARM Limited는 이 설명서 정보의 사용, 정보의 오류나 누락 또는 제품의 잘못된 사용에 따른 어떠한 손실이나 손상도 책임지지 않습니다.

기밀 상태

이 설명서는 기밀 문서가 아닙니다. 이 설명서의 사용, 복사 및 공개에 대한 권한은 ARM과 ARM으로부터 이 설명서를 제공받은 당사자가 동의한 계약 조건에 따라 라이선스 제한의 적용을 받을 수 있습니다.

제품 상태

이 설명서의 정보는 개발된 제품에 대한 최종 정보입니다.

웹 주소

<http://www.arm.com>

목차

RealView Development Suite 시작 설명서

	서문	
	설명서 정보	vi
	사용자 의견	x
1장	소개	
	1.1 RealView Development Suite 구성요소	1-2
	1.2 RealView Development Suite 라이선싱	1-10
	1.3 RealView Development Suite 설명서	1-12
	1.4 RealView Development Suite 예제	1-14
	1.5 ARM Developer Suite	1-16
	1.6 타겟 액세스 지원	1-17
	1.7 RVDS 환경과 관련된 문제 해결	1-19
2장	RVDS 버전 3.0의 기능	
	버전 의 변경 사항	
	RealView Debugger 변경 사항	2-3
	2.3 RealView Compilation Tools 변경 사항	2-4
	2.4 시뮬레이터 지원	2-5
	2.5 CodeWarrior for RVDS 변경 사항	2-6

2.6	설명서 변경 사항	2-7
-----	-----------------	-----

장	RealView Development Suite 시작하기
	빌드 및 디버깅 작업 개요
	예제 프로젝트 사용

부록 A	armenv 도구 사용
	도구
	도구 사용

부록 B	이전 릴리스 정보
B.1	RVDS 버전 2.2에서 RVDS 버전 2.2 SP1로 업그레이드되면서 변경된 사항 B-2
B.2	RVDS 버전 2.1에서 RVDS 버전 2.2로 업그레이드되면서 변경된 사항 B-3
B.3	RVDS 버전 2.0에서 RVDS 버전 2.1로 업그레이드되면서 변경된 사항 B-6
B.4	ADS 버전 1.2.1에서 RVDS 버전 2.2로 업그레이드되면서 변경된 사항 B-8

서문

이 서문에서는 *RealView Development Suite(RVDS)*를 사용하여 소프트웨어 프로젝트를 관리하고 응용 프로그램을 디버깅하는 방법을 보여 주는 [®] *Development Suite 버전 3.0 시작 설명서*에 대해 소개합니다. 이 장에는 다음 단원이 포함되어 있습니다.

- vi페이지의 *설명서 정보*
- x페이지의 *사용자 의견*



설명서 정보

RealView Development Suite에서는 ARM® 아키텍처 기반 프로세서를 타겟으로 하는 소프트웨어 개발 프로젝트를 빌드, 디버깅 및 관리하기 위한 도구를 제공합니다. 이 설명서에는 다음과 같은 내용이 포함되어 있습니다.

- RealView Development Suite를 구성하는 소프트웨어 구성요소에 대한 소개
- RVDS 버전 3.0, 이전 버전의 RVDS 및 ARM Developer Suite™ 버전 1.2의 차이점 요약
- RealView Development Suite를 처음 사용하는 사용자를 위한 용어집

대상 독자

이 설명서는 를 사용하여 아키텍처 기반 프로세서의 개발 프로젝트를 관리하려는 개발자를 위한 것입니다. 여기에서는 사용자가 경험 있는 소프트웨어 개발자이지만 개발 도구에는 익숙하지 않다고 가정합니다.

설명서 사용

이 설명서는 다음 장으로 구성되어 있습니다.

1장 소개

이 장에서는 버전 구성요소 라이선싱 및 설명서에 대해 소개합니다.

2장 RVDS 버전 3.0의 기능

이 장에서는 버전 의 새로운 기능에 대해 설명합니다.

3장 RealView Development Suite 시작하기

이 장에서는 도구를 사용하여 수행할 수 있는 기본 작업을 간략하게 설명합니다. 와 함께 제공되는 예제 프로젝트에 대해서도 설명합니다.

부록 A armenv 도구 사용

이 부록에서는 armenv 도구의 사용 방법을 설명합니다.

부록 B 이전 릴리스 정보

이 장에서는 이전 버전의 와 버전 에 대해 설명합니다.

Typographical conventions

이 설명서에서는 다음과 같은 표기 규칙을 사용합니다

<i>기울임꼴</i>	중요한 사항을 강조 표시하고 특수 용어를 소개하고 내부 상호 참조 및 인용 부분을 나타냅니다
굵은 글꼴	메뉴 이름과 같은 인터페이스 요소를 강조 표시하고 프로세서 신호 이름을 나타냅니다. 적절한 경우 설명 목록의 용어에도 사용됩니다
고정 폭 글꼴	명령 파일 및 프로그램 이름 소스 코드와 같이 키보드로 입력할 수 있는 텍스트를 나타냅니다
<u>고정 폭</u> 글꼴	전체 명령 또는 옵션 대신 사용할 수 있는 약어를 나타냅니다. 밑줄이 그어진 텍스트는 전체 명령이나 옵션 이름 대신 입력할 수 있습니다
고정 폭 <i>기울임꼴</i>	명령 및 함수의 인수를 나타냅니다. 인수는 특정 값으로 대체할 수 있습니다
고정 폭 굵은 글꼴	외부 예제 코드가 사용될 경우 언어 키워드를 나타냅니다
...	경로 이름의 끝에 있는 ...은 필요한 디렉토리가 마지막으로 지정된 디렉토리 이름 아래에 있음을 나타냅니다. 경로 이름이 지정되지 않은 것은 대개 운영 체제마다 차이가 있기 때문입니다. 예를 들면 다음과 같습니다 <i>install_directory\ARM\RVDS\Examples\...</i> 경로 이름의 가운데에 있는 ...은 지정된 디렉토리 이름 사이에 추가 디렉토리가 있음을 나타냅니다. 지정되지 않은 경로 이름은 대개 버전 및 빌드 번호와 플랫폼이 관련된 디렉토리 이름입니다. 예를 들면 다음과 같습니다 <i>install_directory\ARM\RVD\Core\...\etc</i>

추가 정보

이 단원에는 추가 정보를 제공하는 [ARM](#) 와 타사의 게시물 목록이 있습니다

에서는 이 설명서의 내용을 정기적으로 업데이트하고 수정합니다
<http://www.arm.com>에서 정오표 추가 목록 및 ARM FAQ를 참조하십시오.

ARM 게시물

응용 프로그램의 사용을 제어하는 GLOBEtrotter Inc. 제공 FLEX 라이선스 관리 시스템에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- *ARM FLEXlm 라이선스 관리 설명서 버전 4.0 (ARM DUI 0209)*

RVDS 버전 3.0의 라이선스 관리에 대한 자세한 내용은 이 설명서의 버전 4.0을 참조하십시오.

이 설명서는 RVDS 설명서 제품군의 일부입니다. 이 제품군에는 다음 설명서도 포함되어 있습니다.

- *RealView Debugger v3.0 Essentials Guide(ARM DUI 0181)*
 - *RealView Debugger v3.0 User Guide(ARM DUI 0153)*
 - *RealView Debugger v3.0 Target Configuration Guide*
RealView Debugger v3.0 Trace User Guide
RealView Debugger v3.0 RTOS Guide
RealView Debugger v3.0 Command Line Reference Guide(ARM DUI 0175)
 - *RealView Compilation Tools 버전 3.0 핵심 설명서(ARM DUI 0202)*
 - *RealView Compilation Tools 버전 3.0 개발자 설명서(ARM DUI 0203)*
 - *RealView Compilation Tools 버전 3.0 어셈블러 설명서(ARM DUI 0204)*
 - *RealView Compilation Tools 버전 3.0 컴파일러 및 라이브러리 설명서(ARM DUI 0205)*
 - *RealView Compilation Tools 버전 3.0 링커 및 유틸리티 설명서(ARM DUI 0206)*
 - *RealView ARMulator ISS 버전 1.4 사용 설명서*
RealView Development Suite AXD and armsd Debuggers Guide
RealView Development Suite CodeWarrior IDE Guide
RealView Development Suite 용어집
-

다음 설명서에서는 아키텍처 프로세서 관련 장치 및 소프트웨어 인터페이스에 대한 일반적인 정보를 제공합니다

ARM Reference Peripheral Specification

하드웨어 장치에 대한 데이터시트 또는 기술 참조 문서

에서 지원하는 소프트웨어 인터페이스 및 표준에 대한 일반적인 정보는 `install_directory\Documentation\Specifications\`를 참조하십시오

와 함께 사용하기에 적합한 디버그 인터페이스와 관련된 정보는 다음 설명서를 참조하십시오

RealView ICE and RealView Trace User Guide

Multi-ICE[®] User Guide

ARM MultiTrace[™] User Guide

ARM Agilent Debug Interface User Guide

기타 참조 자료

아키텍처에 대한 포괄적인 소개는 다음을 참조하십시오

ARM system-on-chip 구조(Steve Furber 저, 나중화 등 역, 홍릉과학출판사, 2005년, ISBN 8972833592)

CEVA, Inc의 CEVA-Oak, CEVA-TeakLite 및 CEVA-Teak 프로세서에 대한 자세한 내용은 <http://www.ceva-dsp.com>을 참조하십시오.

LSI Logic의 ZSP 부분에 있는 ZSP400 및 ZSP500 프로세서에 대한 자세한 내용은 <http://www.zsp.com>을 참조하십시오.

사용자 의견

RVDS와 해당 설명서에 대한 의견이 있으시면 ARM Limited에 알려 주시기 바랍니다.

RealView Development Suite에 대한 사용자 의견

와 관련된 문제가 있으시면 해당 공급업체에 문의하십시오. 문의 시 다음 사항을 함께 알려 주시면 보다 신속하고 유용한 답변을 받으실 수 있습니다.

사용자 이름 및 회사

제품 일련 번호

사용 중인 릴리스 정보

실행 중인 플랫폼의 세부 사항 예: 하드웨어 플랫폼, 운영 체제 종류 및 버전)

- 문제를 재현하는 작은 독립 실행형 코드 샘플
- 의도한 결과와 실제로 발생한 결과에 대한 명확한 설명
- 사용한 명령(명령 행 옵션 포함)
- 문제를 보여 주는 샘플 출력
- 도구의 버전 문자열(버전 번호 및 날짜 포함)

————— 참고 —————

RealView Debugger와 관련된 문제가 발생하면 **Help** → **Send a Problem Report...** 메뉴를 사용하여 소프트웨어 문제 보고서를 만드십시오. 자세한 내용은 설명서를 참조하십시오.

설명서에 대한 사용자 의견

이 설명서와 관련된 의견이 있으시면 다음 사항을 기재하여 errata@arm.com으로 전자 메일을 보내 주시기 바랍니다.

설명서 제목

설명서 번호

문의 내용과 관련된 페이지 번호

문의 내용에 대한 간략한 설명

추가 및 향상되었으면 하는 기능에 대한 일반적인 제안도 환영합니다.

1장 소개

이 장에서는 *ARMv8-A* 버전 *8.0*에 대해 소개합니다 또한 *ARMv8-A* 버전 *8.0*의 기능을 확장하기 위해 구입할 수 있는 추가 라이선스인 구성요소 응용 프로그램에 대해 설명하고 설명서 제품군에 대해 간략히 설명합니다

이 장에는 다음 단원이 포함되어 있습니다

페이지의	<i>구성요소</i>
페이지의	<i>라이선싱</i>
페이지의	<i>설명서</i>
페이지의	<i>예제</i>
페이지의	
페이지의	<i>타겟 액세스 지원</i>
페이지의	<i>환경과 관련된 문제 해결</i>

1.1 RealView Development Suite 구성요소

이 단원에서는 계열의 프로세서에서 실행되는 임베디드 시스템 응용 프로그램을 개발하기 위한 통합 개발 환경을 제공합니다. 이는 다양한 도구와 지원 설명서 및 예제로 구성된 제품군입니다. 다양한 도구를 사용하여 타겟 하드웨어 또는 소프트웨어 시뮬레이터에서 응용 프로그램을 작성하고 빌드하고 디버깅할 수 있습니다. 이 단원에는 다음과 같은 내용이 포함되어 있습니다.

설치, 예제 및 설명서 디렉토리

페이지의 *CodeWarrior for RVDS*

페이지의 *RealView Compilation Tools*

페이지의 *RealView Debugger*

페이지의 *RealView ARMulator 명령어 세트 시뮬레이터*

페이지의 *명령어 세트 시스템 모델*

페이지의 *RVDS 예제 프로젝트*

페이지의 *ARM eXtended Debugger*

페이지의 *ARM Symbolic Debugger*

1.1.1 RealView Development Suite 설치, 예제 및 설명서 디렉토리

시스템에 설치된 다양한 디렉토리에는 유용한 파일이 들어 있습니다. 설명서에서는 필요할 경우 이러한 디렉토리를 참조합니다.

모든 디렉토리는 주 설치 디렉토리에 있습니다. 또한 이 설명서에 사용된 대부분의 예제는 하나의 예제 디렉토리에 있습니다. 예외가 있을 경우에는 해당 내용이 나오는 부분에서 이를 명시합니다.

표 1-1에서는 주 설치 예제 및 설명서 디렉토리를 보여 줍니다. 여기서 *install_directory*는 기본 설치 디렉토리입니다. 다른 설치 디렉토리를 지정한 경우 경로 이름은 선택한 디렉토리를 기준으로 합니다.

표 1-1 RealView Development Suite 디렉토리

디렉토리	Windows 기본 경로	선 솔라리스 및 레드햇 리눅스 기본 경로
<i>install_directory</i>	C:\Program Files\ARM	~/arm
예제 ^a	<i>install_directory</i> \RVDS\Examples\...	<i>install_directory</i> /RVDS/Examples/...
설명서 ^b	<i>install_directory</i> \Documentation\...	<i>install_directory</i> /Documentation/...

- a. RealView Development Suite와 함께 제공되는 예제에 대한 간략한 설명은 1-14페이지의 *RealView Development Suite 예제*를 참조하십시오
- b. 설명서에 액세스하는 방법은 1-12페이지의 *설명서*를 참조하십시오.

1.1.2 CodeWarrior for RVDS

CodeWarrior for RVDS는 Metrowerks CodeWarrior IDE 버전 5.7을 기반으로 합니다. RVDS 버전 3.0의 CodeWarrior for RVDS는 Windows XP 및 Windows 2000 시스템에서만 지원되며 선 솔라리스 및 레드햇 리눅스용으로는 제공되지 않습니다.

————— 참고 —————

브레이크포인트 설정과 같은 CodeWarrior의 기본 제공 디버깅 도구는 사용하지 말아야 합니다. 응용 프로그램을 컴파일한 후 **Project** → **Debug** 메뉴 옵션을 사용하여 를 시작한 다음 내에서 모든 디버깅 작업을 수행합니다

는 다음과 같은 역할을 합니다

소프트웨어 개발 프로젝트를 관리하기 위한 그래픽 사용자 인터페이스를 제공합니다 를 사용하여 및 프로세서를 타겟으로 하는 C, C++ 및 ARM 어셈블리 언어 코드를 개발할 수 있습니다. 또한 다음과 같은 기능을 제공하여 빌드 주기의 속도를 빠르게 해 줍니다.

- 포괄적인 프로젝트 관리 기능
- 루틴을 빠르게 찾을 수 있도록 도와 주는 코드 탐색 루틴
- ARM 도구에서 프로젝트 코드를 컴파일, 어셈블 및 연결할 수 있도록 구성할 수 있게 해 줍니다.
- 소스 코드 파일, 라이브러리 파일 및 구성 설정을 하나의 *프로젝트*로 구성할 수 있게 해 줍니다. 각 프로젝트를 사용하여 빌드 타겟 설정의 여러 구성을 만들고 관리할 수 있습니다. 예를 들어 디버깅용 빌드 타겟과 릴리스용 빌드 타겟을 컴파일할 수 있으며 코드의 타겟을 ARM7TDMI® 기반의 하드웨어로 지정할 수 있습니다. 빌드 타겟은 동일한 프로젝트의 파일을 공유하면서 동시에 고유한 설정을 사용할 수 있습니다.
- 다음 항목을 제공합니다.
 - 구문 색 지정 기능을 제공하며 CodeWarrior for RVDS 브라우저와 통합되는 소스 코드 편집기
 - 코드에 정의된 심볼의 데이터베이스를 보관하고 소스 코드를 쉽고 빠르게 탐색할 수 있게 해 주는 소스 코드 브라우저

스타일의 정규식을 사용하고 여러 파일에 대해 일괄 검색을 수행할 수 있게 해 주는 검색 및 바꾸기 기능

두 텍스트 파일의 차이점을 찾고 필요할 경우 한 파일의 변경 내용을 다른 텍스트 파일에 병합하고 디렉토리의 내용을 비교할 수 있게 해 주는 파일 비교 기능

를 시작하는 방법은 *RealView Development Suite CodeWarrior IDE Guide*를 참조하십시오

1.1.3 RealView Compilation Tools

를 사용하여 또는 어셈블리 언어 프로그램을 빌드할 수 있습니다 는 다음과 같은 도구로 구성되어 있습니다

armcc 및 Thumb C/C++ 컴파일러

- armasm - ARM 및 Thumb 어셈블러
- armlink - ARM 링커
- armar - ARM 라이브러리 생성기
- fromelf - ARM 이미지 변환 유틸리티
- 지원 라이브러리

RVCT에서 사용할 수 있는 기능에 대한 자세한 내용은 *핵심 설명서*를 참조하십시오.

RVCT 도구 및 유틸리티의 전체 설명과 사용 방법은 RVCT 설명서를 참조하십시오. 이 설명서는 1-12페이지의 *설명서*에 나와 있습니다. 또한 RVCT 도구에 대한 사용 가능한 업데이트 및 패치는 ARM 웹 사이트를 참조하십시오.

1.1.4 RealView Debugger

와 지원되는 디버그 타겟 페이지의 *타겟 액세스 지원* 참조를 함께 사용하면 응용 프로그램을 디버깅하고 프로그램 실행 흐름을 안전하게 제어할 수 있으므로 오류를 빠르게 찾아서 수정할 수 있습니다

참고

선 솔라리스 또는 레드햇 리눅스에서 사용하는 것과 관련된 자세한 내용은 선 솔라리스 및 레드햇 리눅스에서의 *설명서*를 설명하는 부록을 참조하십시오 이 부록은 에 있습니다

에서는 다음을 지원합니다

다중 프로세서 디버깅 *다중 프로세서 디버깅* 참조

디버깅 페이지의 *디버깅* 참조

트레이스 분석 및 프로파일링 페이지의 *트레이스 분석 및 프로파일링* 참조

운영 체제 인식 기능 페이지의 *인식* 참조 이 기능은 공급업체별 플러그인을 다운로드하면 사용할 수 있습니다

의 기본 라이선스를 사용하면 단일 아키텍처 기반 프로세서에서 실행되는 응용 프로그램을 디버깅할 수 있습니다 그러나 추가 라이선스를 구입하면 기능을 확장하여 다중 프로세서에서 실행되는 응용 프로그램을 디버깅하고 에서의 디버깅을 지원할 수 있습니다 자세한 내용은 페이지의 *라이선싱*을 참조하십시오

에서 사용할 수 있는 기능에 대한 자세한 내용을 참조하십시오

의 전체 설명과 사용 방법은 설명서를 참조하십시오 이 설명서는 페이지의 *설명서*에 나와 있습니다

다중 프로세서 디버깅

다중 프로세서 디버깅을 사용하면 둘 이상의 프로세서에서 실행되는 소프트웨어 시스템을 디버깅할 수 있습니다 프로세서는 단일 개발 보드에 있을 수도 있고 다중 개발 보드에 있을 수도 있습니다 두 경우 모두 는 각 프로세서에 대해 서로 다른 연결을 사용합니다

다중 프로세서 디버깅을 사용하면 혼합 코어 시스템을 디버깅하고 프로세서 연산을 동기화할 수 있습니다

다중 보드 연결에 동일한 프로세서를 사용하는 경우에는 새 타겟 설명을 만들어야 합니다 사용자 지정 타겟을 만드는 방법은 를 참조하십시오

다중 프로세서 디버깅을 사용하려면 라이선스를 별도로 구입해야 합니다 자세한 내용은 페이지의 *라이선싱*을 참조하십시오

다중 프로세서 디버깅에 대한 자세한 내용은 에서 다중 타겟 디버깅을 설명하는 장을 참조하십시오



DSP 디버깅

는 다음 의 디버깅을 지원합니다

수정 버전 및 및

수정 버전 및

삼성 의

및

디버깅을 사용하려면 라이선스를 별도로 구입해야 합니다 자세한 내용은 페이지의 *RealView Development Suite 라이선싱*을 참조하십시오

지원에 대한 자세한 내용은 *RealView Debugger User Guide*에서 지원을 설명하는 장을 참조하십시오

트레이스, 분석 및 프로파일링

는 트레이스 하드웨어나 하드웨어 시뮬레이터를 사용한 트레이싱을 지원합니다 사용할 수 있는 트레이스 하드웨어는 다음과 같습니다

Embedded Trace Macrocell™ 이 포함된 프로세서

™(ETB™)와 같은 칩 내부 트레이스 버퍼

- RealView ICE와 같은 JTAG(Joint Test Action Group) 인터페이스 장치

ETM이 아닌 기본 트레이스 지원은 RealView ARMulator® ISS 하드웨어 시뮬레이터에서 제공합니다.

RealView Debugger에서 사용할 수 있는 트레이스 기능에 대한 자세한 내용은 *Debugger Extensions User Guide*에서 트레이싱을 설명하는 장을 참조하십시오.

OS 인식

인식은 에 기본적으로 제공된 확장 기능입니다 이 확장을 사용하려면 먼저 사용 중인 용 플러그인을 얻어야 합니다 플러그인이 있으면 에서는 또는 어셈블러 소스 수준 심볼 디버그 정보 외에 스레드 및 리소스 예 큐 사서함 및 세마포어 도 참조합니다 용 플러그인을 얻는 방법은 페이지의 다운로드를 참조하십시오

의 인식 기능은 공급업체에 관계없이 제공되며 둘 이상의 공급업체에서 플러그인을 다운로드하여 사용할 수 있습니다 따라서 다양한 플랫폼용 응용 프로그램을 개발할 수 있습니다 둘 이상의 공급업체에서 다운로드한 플러그인을 동일한 디버깅 세션에서 사용하려면 새 타겟 설명을 만들어야 합니다 사용자 지정 타겟을 만드는 방법은 를 참조하십시오

지원에 대한 자세한 내용은 에서 지원을 설명하는 장을 참조하십시오 또한 사용 가능한 지원에 대한 자세한 내용은 웹 사이트를 참조하십시오

다운로드

메뉴에서 다양한 다운로드에 액세스할 수 있습니다

→ ARM on the Web → Goto RTOS Awareness Downloads

웹 사이트의 웹 페이지를 표시합니다 이 페이지에서 필요한 플러그인을 찾아 다운로드할 수 있습니다

Help → ARM on the Web → Goto Update and Utility Downloads

웹 사이트의 *RealView Debugger - Updates & Utilities* 웹 페이지를 표시합니다 이 페이지에서 소프트웨어 업데이트와 유틸리티를 찾아 다운로드할 수 있습니다

1.1.5 RealView ARMulator 명령어 세트 시뮬레이터

*명령어 세트 시뮬레이터*는 프로세서의 명령어 세트 및 아키텍처와 메모리 시스템과 주변 장치를 함께 시뮬레이션합니다

를 사용하면 타겟 하드웨어 없이도 임베디드 응용 프로그램을 개발하고 디버깅할 수 있습니다 는 하드웨어가 아직 개발 중이거나 사용할 수 있는 개발 보드의 수가 제한되어 있는 경우에 유용합니다

표 1-2에서는 Windows, 선 솔라리스 및 레드 햇 리눅스의 ARM 디버거에서 사용할 수 있는 RVISS 인터페이스 연결을 보여 줍니다. RealView Connection Broker를 통해 RVISS에 연결할 경우 RVISS의 여러 인스턴스에 연결하여 다중 프로세서 시스템을 시뮬레이션할 수 있습니다.

표 1-2 각 플랫폼에서 지원되는 RealView ARMulator ISS 연결

ARM 디버거	Remote Debug Interface 연결	RealView Connection Broker 연결
RealView Debugger	Windows ^a	Windows, 선 솔라리스 및 레드 햇 리눅스
AXD	Windows	사용할 수 없음
armsd	Windows, 선 솔라리스 및 레드 햇 리눅스	사용할 수 없음

- a. 이번 릴리스에서 RealView Debugger의 RVISS에 대한 원격 디버그 인터페이스 연결(RDI)은 향후 제공되지 않을 예정입니다.

RVISS에서 사용할 수 있는 기능에 대한 자세한 내용은
를 참조하십시오.

1.1.6 명령어 세트 시스템 모델

명령어 세트 시스템 모델은 및 프로세서의 명령어 세트와 아키텍처를 시뮬레이션합니다 자세한 내용은
를 참조하십시오

1.1.7 RVDS 예제 프로젝트

에는 예제 프로젝트가 제공됩니다 자세한 내용은 페이지의
예제를 참조하십시오 이러한 예제 프로젝트는 다음 디렉토리에
있습니다

`install_directory\RVDS\Examples\...`

의 시작 메뉴에서 예제 프로젝트 디렉토리를 탐색하려면 다음을 선택합니다

프로그램 → → **RealView Development Suite v3.0** → **Examples**

1.1.8 ARM eXtended Debugger

는 단일 프로세서 디버거로 용으로만 제공됩니
다 와 지원되는 디버그 타겟 페이지의 *타겟 액세스 지원* 참조 을 함께 사용
하면 응용 프로그램을 디버깅하고 프로그램 실행 흐름을 완전하게 제어할 수 있으므
로 오류를 빠르게 찾아서 수정할 수 있습니다

참고

는 레거시 및 지원을 위해 제공되며 장치에서
는 작동하지 않습니다

를 사용하는 방법은 를
참조하십시오

1.1.9 ARM Symbolic Debugger

는 명령 행 인터페이스에서 실행되는 단일 프로세서
디버거로 선 솔라리스 및 레드햇 리눅스에서 사용할 수 있습니다

를 사용하면 의 를 통해 응용 프로그램을 디버
깅할 수 있습니다 자세한 내용은 페이지의 *명령어 세트 시뮬레
이터*를 참조하십시오

참고

가 포함된 는 및 타겟만 지원합니다

를 사용하는 방법은
를 참조하십시오

1.2 RealView Development Suite 라이선싱

의 모든 라이선싱은 라이선스 관리 시스템에서 제어됩니다 라이선스
센스를 추적하고 제어하려면 서버 소프트웨어를 사용합니다 이제
<http://license.arm.com>의 페이지를 사용하여 라이선스를 요청할
수 있습니다 자세한 내용은 *라이선스 관리 설명서*를 참조하십시오

참고

에 직접 문의하여 라이선스를 요청할 필요가 없습니다

기능에 별도로 사용할 수 있는 라이선스는 다음과 같습니다

다중 프로세서 디버깅 라이선스

CEVA-Oak 및 CEVA-Teaklite DSP 디버깅 라이선스

CEVA-Teak DSP 디버깅 라이선스

페이지의 *LSI Logic ZSP DSP 디버깅 라이선스*

다중 프로세서 디버깅 라이선스

다중 프로세서 디버깅 라이선스를 사용하면 둘 이상의 프로세서에서 실행되는 소프트웨어 시스템을 디버깅할 수 있습니다 자세한 내용은 페이지의 *다중 프로세서 디버깅*을 참조하십시오

CEVA-Oak 및 CEVA-Teaklite DSP 디버깅 라이선스

및 지원 라이선스를 사용하면
및 에서 실행되는 응용 프로그램을 디버깅할 수 있습니다
다 자세한 내용은 페이지의 *디버깅*을 참조하십시오

1.2.3 CEVA-Teak DSP 디버깅 라이선스

지원 라이선스를 사용하면 에
서 실행되는 응용 프로그램을 디버깅할 수 있습니다 자세한 내용은 페이지의
*디버깅*을 참조하십시오

1.2.4 LSI Logic ZSP DSP 디버깅 라이선스

지원 라이선스를 사용하면 및
에서 실행되는 응용 프로그램을 디버깅할 수 있습니다. 자세한 내용은
페이지의 *디버깅*을 참조하십시오.



1.3 RealView Development Suite 설명서

이 단원에서는 와 함께 제공되는 설명서에 대해 설명합니다 이 장에는 다음 단원이 포함되어 있습니다

설명서 목록

온라인 추가 정보 얻기

1.3.1 설명서 목록

설명서는 다음과 같이 구성되어 있습니다

시작 설명서 이 설명서

라이선스 관리 설명서

핵심 설명서

개발자 설명서

어셈블러 설명서

컴파일러 및 라이브러리 설명서

링커 및 유틸리티 설명서

용어집

및 타사의 관련 게시물은 각 설명서의 추가 정보 단원을 참조하십시오

1.3.2 온라인 추가 정보 얻기

설치된 플랫폼에 따라 및 선 솔라리스용 및 형식과 레드 핫 리눅스용 형식의 전체 설명서 제품군을 온라인으로 볼 수 있습니다 및 파일에는 동일한 정보가 들어 있습니다 설명서는 페이지의 표 에 표시된 설명서 디렉토리에 설치됩니다

버전에는 단일인 [Collection.pdf](#)에서 액세스할 수 있는 새로운 설명서 제품군이 포함되어 있습니다. 전체 설명서 제품군을 설치할 경우 설명서 모음에서 모든 파일의 텍스트를 검색할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

의
[선 솔라리스 및 레드햇 리눅스의](#)

Windows의 RVDS

및 선 솔라리스 시스템에서는 전자 설명서 형식의 설명서를 온라인으로 볼 수 있습니다. 설명서의 내용과 설명서의 내용은 동일합니다.

의 시작 메뉴에서 프로그램 → 을 선택합니다. 그런 후 다음 중 하나를 선택합니다.

파일을 보려면 →
를 선택합니다.

파일을 보려면 을 선택합니다.

선 솔라리스 및 레드햇 리눅스의 RVDS

전체 설명서 제품군은 선 솔라리스용 및 파일과 레드햇 리눅스용 파일로 사용할 수 있습니다. 데스크톱 링크를 설정한 경우 해당 링크를 사용하여 필요한 설명서에 액세스합니다.

참고

파일은 레드햇 리눅스에는 설치되지 않습니다.

1.4 RealView Development Suite 예제

설명서에 나오는 대부분의 예제 코드는 주 예제 디렉토리에 있습니다 자세한 내용은 페이지의 [설치 예제 및 설명서 디렉토리](#)를 참조하십시오

이 디렉토리에는 이 설명서에서 설명되지 않은 예제 코드도 들어 있습니다 자세한 내용은 각 예제 디렉토리의 `readme.txt`를 참조하십시오 예제는 다음 하위 디렉토리에 설치됩니다

asm	어셈블리 언어 프로그래밍에 대한 몇 개의 예제가 있습니다 이러한 예제는 어셈블러 설명서 에서 사용됩니다
cached_dhry	예제를 기반으로 빌드된 캐시 및 초기화 루틴의 예제가 있습니다
Cortex-M3	예제 스캐터 파일 및 빌드 스크립트가 포함된 프로세서용 Hello World 예제가 있습니다.
cpp	몇 개의 기본 C++ 예제가 있습니다.
databort	표준 데이터 어보트 처리기의 디자인 설명서와 예제 코드가 있습니다.
dcc	디버그 통신 채널의 사용 방법을 보여 주는 예제 코드가 있습니다. 이 예제는 개발자 설명서 에 설명되어 있습니다.
dhystone	Dhystone 벤치마크 예제입니다. 이 예제는 RealView Debugger 설명서에서 사용됩니다.
dsp	dspfns.h 헤더 파일 전체를 사용하는 데 필요한 작은 소스 파일이 있습니다. 이 파일은 DSP 유형의 기본 연산 세트를 정의하고 ARM 컴파일러의 인라인 어셈블리 기능을 사용하는 방법을 보여 줍니다.
emb_sw_dev	개발자 설명서 의 임베디드 소프트웨어 개발을 설명하는 장에서 참조되는 예제 프로젝트가 있습니다. 다음과 같은 하위 디렉토리가 있습니다.
buildn	예제 프로젝트를 빌드하기 위한 배치 파일과 메이크파일이 있습니다. 각 프로젝트에 대한 설명은 관련 <code>readme.txt</code> 파일을 참조하십시오.
dhry	Dhystone 벤치마킹 프로그램의 소스 파일이 있습니다. 이 프로그램은 개별 <code>buildn</code> 디렉토리의 예제 프로젝트에 대한 코드 베이스를 제공합니다.

<code>include</code>	사용자 정의 헤더 파일이 있습니다.
<code>scatter</code>	예제 프로젝트를 빌드하는 데 사용되는 스캐터 파일이 있습니다.
<code>source</code>	예제 프로젝트를 빌드하는 데 필요한 그 밖의 모든 소스 파일이 있습니다.
<code>fft_v5te</code>	ARMv5TE(ARM 아키텍처 v5TE)에 대해 최적화된 고속 푸리에 변환 코드가 있습니다.
<code>inline</code>	ARM C 및 C++ 코드를 컴파일할 때 인라인 어셈블리를 사용하는 방법을 보여 주는 예제가 있습니다. 자세한 내용은 개발자 설명서 에서 혼합 C, C++ 및 어셈블리 언어를 설명하는 장을 참조하십시오.
<code>interwork</code>	ARM 코드와 Thumb 코드 간의 인터워킹 방식을 보여 주는 예제가 있습니다. 자세한 내용은 개발자 설명서 에서 ARM 과 Thumb의 인터워킹을 설명하는 장을 참조하십시오.
<code>mmugen</code>	MMUgen 유틸리티의 소스와 설명서가 있습니다. 이 유틸리티는 필요한 가상 대 실제 주소 변환을 설명하는 규칙 파일에서 MMU 페이지 테이블 데이터를 생성할 수 있습니다.
<code>picpid</code>	위치 독립적인 코드를 작성하는 방법을 보여 주는 예제가 있습니다.
<code>sorts</code>	ARM C 라이브러리에 사용되는 삽입 정렬, 셸 정렬 및 빠른 정렬을 비교하는 예제 코드가 있습니다.
<code>svc</code>	예제 SVC(관리자 호출) 처리기가 있습니다.
<code>unicode</code>	다중 바이트 문자 지원을 평가할 수 있게 해 주는 예제 코드가 있습니다.
<code>vfpsupport</code>	VFP(벡터 부동 소수점) 연산을 설정하고 실행하기 위한 예제 코드가 있습니다. 또한 VFP를 사용할 때 디버그 시스템을 구성하기 위한 다양한 유틸리티 파일과 PDF 형식의 <i>Application Note 133 Using VFP in RVDS</i> 가 들어 있습니다.

1.5 ARM Developer Suite

에는 버전 의 전체 버전도 포함되어 있습니다
버전 은 를 설치할 때 함께 설치되지 않습니다 버전 을 사용
해야 하는 경우에는 별도로 설치해야 합니다

1.5.1 ADS와 RVDS를 모두 설치할 때의 유의 사항

와 를 모두 설치할 때는 다음 사항에 주의하십시오
뿐만 아니라 도 설치할 수 있지만 와 함께 설치된
를 모두 사용하는 경우에는 매우 조심해야 합니다 자세한
한 내용은 [칼리스 정보를 참조하십시오](#) 그러나
를 설치한 경우에는 디버거를 사용하여 에 연결할
수 있습니다 자세한 내용은 페이지의 [명령어 세트 시물
레이터](#)를 참조하십시오
시스템에 와 를 모두 설치하려는 경우에는
유틸리티를 사용해야 합니다 그러나 유틸리티를 사용하여 설치된
다른 버전의 로 전환할 수는 없습니다 는 웹 사이트에서
무료로 다운로드할 수 있습니다

1.5.2 이전 버전의 ADS 컴파일 도구 설치

이전 버전의 가 필요할 경우를 위해 버전 과 동일한 에 이전 버
전의 가 포함되어 있습니다
의 경우 버전 및 버전 의 설치 관리자가 제공됩니다
선 솔라리스 및 레드햇 리눅스의 경우 버전 및 버전 의 컴파일 도구만 제
공됩니다 해당 설치 관리자는 포함되어 있지 않으므로 파일을 수동으로 복사해야 합
니다 이 파일은 의 `ads_1_1` 및 `ads_1_0_1` 디렉토리에 있습니다 다음을 수행해
야 합니다

플랫폼에 필요한 구성요소와 이를 복사할 위치를 결정합니다

플랫폼에 종속적인 디렉토리에서 올바른 실행 파일을 선택합니다

필요한 라이브러리를 복사합니다

경로에 실행 파일의 위치가 포함되어 있고 다음 환경 변수가 설정되어 있는지
확인합니다

`ARMINC` 포함 파일이 들어 있는 디렉토리를 지정합니다

`ARMLIB` 라이브러리 파일이 들어 있는 디렉토리를 지정합니다

1.6 타겟 액세스 지원

다음 단원에서는 RVDS 버전 3.0 및 ADS 버전 1.2.1의 ARM 디버거에서 지원하는 타겟 액세스를 간략히 설명합니다.

- [에서의 타겟 액세스 지원](#)
- 1-18페이지의 [선 솔라리스 및 레드햇 리눅스에서의 타겟 액세스 지원](#)
- 1-18페이지의 [버전 1.2.1 타겟 매체 지원](#)

에서의 타겟 액세스 지원

의 경우 의 디버거에서 지원하는 타겟 액세스는 표 과 같습니다

표 1-3 Windows에서 지원되는 타겟 액세스

타겟 액세스	버전 3.0 이상	AXD 버전 1.3.1	armsd
ADI(Agilent Debug Interface)	예	예	
Angel 디버그 모니터(Remote_A)	예	예	예
ISSM(명령어 세트 시스템 모델)	예		
Multi-ICE	예	예	
MultiTrace™	예		
RealMonitor	예	예	
RealView ARMulator ISS	RealView Simulator Broker(localhost)만 해당	RDI만 해당	RDI만 해당
RealView ICE	예		
RealView Trace	예		
Versatile 플랫폼의 기본 제공 RealView ICE Micro Edition(USB 연결)	예		

다음 사항에 주의하십시오

플랫폼에 대한 연결을 사용하려면 사용자 지정 설치를 수행하고 디버거 지원 옵션을 선택해야 합니다

소프트웨어는 와 함께 자동으로 설치됩니다

- 이제 RealView Debugger에서 RealMonitor와 RealView ICE를 함께 사용할 수 있습니다. RealMonitor와 RealView ICE를 함께 사용하는 것에 대한 자세한 내용은 [여기](#)를 참조하십시오.
- MultiTrace 지원을 사용하려면 [여기](#) (TDT)를 설치해야 합니다.
- 설치한 후에는 RealView Debugger의 ARM-A-RR 타겟 목록에 ADI DLL(gateway.dll)을 추가해야 합니다. 자세한 방법은 [여기](#)에서 RDI 타겟 사용에 대한 설명을 참조하십시오.

선 솔라리스 및 레드햇 리눅스에서의 타겟 액세스 지원

선 솔라리스 및 레드햇 리눅스의 경우 RVDS의 ARM 디버거에서 지원하는 타겟 액세스는 표 1-4와 같습니다.

표 1-4 선 솔라리스 및 레드햇 리눅스에서 지원되는 타겟 매체

타겟 액세스	RealView Debugger 버전 3.0 이상	armsd
ISSM(MxDI)	예	
RealView ARMulator ISS	RealView Simulator Broker(localhost)만 해당	RDI만 해당
RealMonitor	예	
RealView ICE	예	

이제 RealView Debugger에서 RealMonitor와 RealView ICE를 함께 사용할 수 있습니다. RealMonitor와 RealView ICE를 함께 사용하는 것에 대한 자세한 내용은 [여기](#)를 참조하십시오.

버전 타겟 매체 지원

ADS 버전 1.2.1의 AXD 디버거에서는 다음과 같은 타겟 매체를 지원합니다.

- RealView ARMulator ISS(RDI 인터페이스를 사용하는 경우만 해당)
- RealMonitor
- Multi-ICE
- MultiTrace(TDT를 설치해야 함)
- Agilent Debug Interface(Gateway)
- Angel 디버그 모니터(Remote_A)

1.7 RVDS 환경과 관련된 문제 해결

RVDS의 구성요소 응용 프로그램을 실행하는 데 문제가 있으면 RVDS 환경이 올바르게 구성되어 있는지 확인합니다.

- 선 솔라리스 또는 레드햇 리눅스의 경우 셸에 따라 `RVDS30env.sh` 스크립트나 `RVDS30env.csh` 스크립트를 실행합니다. 이 방법은 선 솔라리스 또는 레드햇 리눅스에서 RVDS 환경을 설정하는 데 사용되는 기본 방법입니다. 이 스크립트의 실행에 대한 자세한 내용은 [설치 설명서](#)를 참조하십시오.
- Windows에 RVDS를 설치할 경우 설치 중 고급 설정 옵션에서 옵션을 선택 취소하지 않았으면 설치 과정에서 `armenv` 유틸리티가 자동으로 실행됩니다. 그러나 설치 후 `armenv` 유틸리티를 사용하여 RVDS 환경을 수정할 수도 있습니다. `armenv` 유틸리티의 사용 방법은 [부록 A 도구 사용](#)을 참조하십시오. 이 유틸리티는 선 솔라리스 및 레드햇 리눅스 시스템에서도 사용할 수 있습니다.

참고

이번 릴리스에서는 사용자 지정 설치에 `armenv` 유틸리티를 사용할 수 없습니다. Windows에서 사용자 지정 설치를 수행한 경우에는 환경 변수를 사용자가 직접 설정해야 합니다. 자세한 내용은 1-20페이지의 [주요 환경 변수](#)를 참조하십시오. 선 솔라리스 또는 레드햇 리눅스에서는 `RVDS30env.sh` 스크립트나 `RVDS30env.csh` 스크립트를 사용합니다.

주요 환경 변수

표 1-5에서는 설정해야 하는 주요 환경 변수를 보여 줍니다 여기서 ...은 사용자 설치 환경의 해당 경로 요소로 바꾸십시오 이러한 환경 변수를 설정하려면 페이지의 *RVDS 환경과 관련된 문제 해결*에 설명된 기본 방법을 사용합니다 또한 PATH 환경 변수에 다양한 구성요소 응용 프로그램 실행 파일의 위치가 포함되어 있는지 확인합니다

표 1-5 Windows의 주요 RVDS 환경 변수

환경 변수	설정
ARMROOT	설치 디렉토리 루트 <i>install_directory</i> 를 지정합니다. Windows에서 기본 설치 디렉토리 루트는 다음과 같습니다. C:\Program Files\ARM
ARMCONF	RVISS 및 여러 RDI 타겟 구성 파일의 위치를 지정하는 데 사용됩니다. <i>install_directory</i> \RDI\armperip\...\...; <i>install_directory</i> \RVARMulator\v6ARMulator\...\...\win_32-pentium; <i>install_directory</i> \RVARMulator\ARMulator\...\...\win_32-pentium
ARMDLL	RVISS 및 여러 RDI 타겟 DLL 파일의 위치를 지정하는 데 사용됩니다. <i>install_directory</i> \RVARMulator\v6ARMulator\...\...\win_32-pentium; <i>install_directory</i> \RVARMulator\ARMulator\...\...\win_32-pentium; <i>install_directory</i> \ARM\RDI\Targets\Remote_A\...\...\win_32-pentium; <i>install_directory</i> \RDI\rdimsvr\...\...\win_32-pentium
ARLMD_LICENSE_FILE	RVDS 라이선스 파일의 위치를 지정합니다. 이 환경 변수에 대한 자세한 내용은 <i>ARM FLEXlm 라이선스 관리 설명서</i> 를 참조하십시오
RVCT30BIN	프로그램 실행 파일을 지정합니다 <i>install_directory</i> \RVCT\Programs\...\...\win_32-pentium
RVCT30INC	컴파일러 포함 파일을 지정합니다 <i>install_directory</i> \RVCT\Data\...\...\include\windows
RVCT30LIB	컴파일러 포함 파일을 지정합니다 <i>install_directory</i> \RVCT\Data\...\...\lib
RVDEBUG_HLPPATH	온라인 도움말 파일을 지정합니다 <i>install_directory</i> \Documentation\RVD\...\release\windows\onlinehelp
RVDEBUG_INSTALL	실행 파일을 지정합니다 <i>install_directory</i> \RVD\Core\...\...\win_32-pentium

2장

RVDS 버전 3.0의 기능

이 장에서는 *Development Suite* 버전 의 새로운 기능을 설명하고 버전 에서 변경된 주요 사항을 요약합니다 이 장에는 다음 단원이 포함되어 있습니다

페이지의 *RVDS 버전 3.0의 변경 사항*

페이지의 *RealView Debugger 변경 사항*

페이지의 *RealView Compilation Tools 변경 사항*

페이지의 *시뮬레이터 지원*

페이지의 *CodeWarrior for RVDS 변경 사항*

페이지의 *설명서 변경 사항*

버전 3.0의 변경 사항

버전 3.0의 주요 변경 사항은 다음 단원에서 설명합니다

버전 3.0의 새로운 기능

향후 제공되지 않을 기능과 제거된 기능

2.1.1 RVDS 버전 3.0의 새로운 기능

버전 3.0에서는 다음과 같은 새로운 기능을 사용할 수 있습니다

◎ 아키텍처 지원

- ARMv7용 (Thumb-2EE) 지원
- ARM Cortex™ 프로세서 계열 지원
 - Cortex-A8
 - Cortex-M3
- Cortex-A8 및 Cortex-M3 프로세서의 시뮬레이터 모델을 사용할 수 있습니다. 이러한 모델은 RealView Debugger의 새 ISSM(명령어 세트 시스템 모델) 타겟 액세스를 통해 액세스할 수 있습니다.
- 리눅스 응용 프로그램 디버깅에 대한 지원이 추가되었습니다.

향후 제공되지 않을 기능과 제거된 기능

RVDS 버전 3.0에서 다음 기능은 향후 제공되지 않을 예정이거나 제거되었습니다.

- (AXD) 및 (armsd)에 대한 지원은 향후 제공되지 않을 예정입니다.
- CodeWarrior의 마이크로파일 임포터와 Batch File Runner 기능은 향후 제공되지 않을 예정입니다.
- Multi-ICE® 직접 연결을 통한 원격 RealView Debugger 연결에 대한 지원이 제거되었습니다. 따라서 DSP 프로세서에 대한 연결은 별도로 구입해야 하는 RealView ICE를 통해서만 사용할 수 있습니다.
- RealView Debugger 프로젝트 관리자 및 관련 기능이 제거되었습니다.

향후 제공되지 않을 다른 기능에 대한 자세한 내용은 다음을 참조하십시오.

- *핵심 설명서*
-

RealView Debugger 변경 사항

버전 의 주요 변경 사항은 다음과 같습니다

는 이제 단일 프로세스로 실행됩니다 타겟 매체 서버 는 더 이상 개별 엔터티로 존재하지 않습니다

Connection Control 창이 다시 엔지니어링되었습니다 자세한 내용은 를 참조하십시오

Synch 탭의 기능은 이제 별도의 **Synchronization Control** 창에서 사용할 수 있습니다 자세한 내용은 를 참조하십시오

Register 창이 다시 엔지니어링되었습니다 이제 선택한 레지스터를 **User** 탭에 복사하여 사용자별 뷰를 만들 수 있습니다 자세한 내용은 를 참조하십시오

리눅스 응용 프로그램 디버깅에 대한 지원이 추가되었습니다 자세한 내용은 를 참조하십시오

프로젝트 관리자와 관련 기능이 제거되었으므로 더 이상 내에서 프로젝트를 만들고 이미지를 빌드할 수 없습니다 그러나 소스 코드 편집 및 검색 기능은 여전히 사용할 수 있습니다

참고

버전 에서 프로젝트를 만들고 빌드하려면 를 사용합니다 자세한 내용은 페이지의 를 참조하십시오

시뮬레이터 지원이 변경되었습니다 자세한 내용은 페이지의 *시뮬레이터 지원*을 참조하십시오

가 다시 엔지니어링되었습니다 로컬 호스트 연결의 경우 가 여전히 를 실행하지만 원격 시뮬레이터 연결에 경우에는 시작 방법이 변경되었습니다 이제 원격 워크스테이션에서 를 시작하려면 사용자 이름을 지정해야 합니다 자세한 내용은 를 참조하십시오

참고

직접 연결에 대한 지원이 버전 에서 제거되었습니다

의 변경 사항에 대한 자세한 내용은 를 참조하십시오

RealView Compilation Tools 변경 사항

버전 의 주요 변경 사항은 다음과 같습니다

버전 에서는 용 를 지원합니다

어셈블러는 프로세서용 코드를 개발하기 위해 인텔 무선 기술 명령어를 어셈블링하는 데 사용할 수 있습니다

버전 에서는 *ABI for the ARM Architecture(기본 표준)[BSABI]*에 기술된 대로 DWARF 3(초안 표준 9.6) 디버그 테이블을 완전하게 지원합니다.

- ARM 컴파일러 및 링커는 프로그램에서 다중 스레드를 사용할 수 있게 해 주는 TLS()를 지원합니다.
- ARM 컴파일러는 개선된 루프 최적화를 지원합니다.

RVCT의 변경 사항에 대한 자세한 내용은
참조하십시오.

핵심 설명서를

시뮬레이터 지원

RVDS에서는 현재 다음과 같은 시뮬레이터 지원을 제공합니다.

- Cortex-A8 및 Cortex-M3 프로세서를 시뮬레이션하는 ISSM
- 이제 RealView ARMulator® ISS(RVISS)에서 MPCore™로 시뮬레이션된 타겟을 사용할 수 있습니다. 그러나 이 타겟은 다중 프로세서를 모델링하지 않으므로 이 모델에 연결할 때는 단일 프로세서에만 연결됩니다.

RDI ARMulator로 시뮬레이션된 타겟은 더 이상 사용할 수 없습니다. 다음 중 하나를 사용하십시오.

- RVISS를 사용하여 시뮬레이션된 ARM® 프로세서에 연결하려면 localhost의 new_arm 연결 타겟 액세스를 사용합니다.
- Cortex 모델 중 하나에 연결하려면 ISSM 타겟 액세스를 사용합니다.

두 타겟 액세스는 모두 RealView Debugger와 함께 설치됩니다.

CodeWarrior for RVDS 변경 사항

의 주요 변경 사항은 다음과 같습니다

이제 **External Build Wizard**가 지원됩니다. 이는 향후 제공되지 않을 매크로파일 임포터와 **Batch File Runner** 기능을 대체하기 위한 것입니다.

- .cc 파일 확장명에 대한 지원이 추가되었습니다.
- 이제 .cmd와 같이 인식되지 않는 소스 파일 확장명을 사용하면 CodeWarrior에서 경고를 표시합니다.
- **New** 대화 상자에서 다음 항목이 지원되지 않습니다

Project 탭의 PowerParts Application Wizard 및 PowerParts Component Wizard 항목

File 탭의 Component Catalog File 및 PowerParts Palette Wizard 항목

Object 탭의 PowerParts Form Wizard 항목

컴파일 도구가 변경되면서 몇 가지 패널 설정이 추가되거나 제거되었습니다
자세한 내용은 [핵심 설명서](#)를 참조하십시오

자세한 내용은 를 참조하십시오

설명서 변경 사항

설명서에는 의 새 기능뿐 아니라 설명서의 주요 변경 사항에 대해서도 설명되어 있습니다 설명서는 다음과 같이 재구성되었습니다

의 정보는 다음 설명서에 포함되었습니다

지원에 대해 설명하는 장이 에 포함되었습니다

다중 타겟 디버깅에 대해 설명하는 장이 에 포함되었습니다

의 트레이싱에 대해 설명하는 장이 에 포함되었습니다

지원에 대해 설명하는 장이 설명서에 포함되었습니다

설명하는 장이 에 포함되었던 타겟 연결에 대해 에 포함되었습니다

가 보다 효율적인 작업 기반 설명서로 다시 구성되었습니다

의 프로젝트 관리자 기능이 제거되었으므로 *Debugger Project Management Guide*는 더 이상 제공되지 않습니다

설명서에 대한 그 밖의 자세한 변경 사항은 다음을 참조하십시오

RealView Debugger Essentials Guide

RealView Compilation Tools 핵심 설명서

3장

RealView Development Suite 시작하기

와 함께 제공되는 구성요소 제품을 사용하여 응용 프로그램을 구성하는 하나 이상의 이미지를 빌드하고 디버깅할 수 있습니다 이 장에서는 도구를 사용하여 빌드 및 디버깅하기 위한 기본 작업에 대해 설명합니다 이 장에는 다음 단원이 포함되어 있습니다

 페이지의 *빌드 및 디버깅 작업 개요*

 페이지의 *예제 프로젝트 사용*

빌드 및 디버깅 작업 개요

표 3-1에서는 도구를 사용하여 응용 프로그램을 빌드 및 디버깅하기 위한 기본 작업을 상위 수준 절차로 보여 주고 세부 정보가 있는 참조 설명서 제목을 제공합니다.

참조 설명서에 설명된 작업은 표 3-1에 설명된 순서와 같지 않을 수도 있습니다. 도구를 처음 사용하는 경우에는 참조 설명서에 설명된 순서에 따라 차례로 작업하는 것이 좋습니다. 표 3-1에 표시된 순서는 일반적으로 수행되는 작업 순서를 반영한 것입니다.

표 3-1 기본 빌드 및 디버깅 작업

단계	설명	참조
1	<p>프로젝트를 관리하고 빌드하는 데 사용할 RVDS 응용 프로그램을 선택합니다.</p> <ul style="list-style-type: none"> CodeWarrior for RVDS를 사용하려면 3단계로 진행합니다. RVCT를 사용하여 명령 행에서 빌드하려면 2단계로 진행합니다. 	
2	<p>RVCT 빌드 도구를 직접 사용하려면 필요한 빌드 명령이 들어 있는 매크로파일이나 Windows 명령 파일을 만듭니다.</p> <p>9단계로 진행하여 RealView Debugger에서 이미지를 로드하고 디버깅합니다.</p>	<p><i>RealView Compilation Tools 핵심 설명서</i></p>
	<p>를 시작합니다</p>	<p><i>RealView Development Suite CodeWarrior IDE Guide</i></p>
4	<p>CodeWarrior for RVDS 프로젝트가 이미 있으면 6단계로 진행합니다.</p> <p>그렇지 않으면 응용 프로그램의 CodeWarrior for RVDS 프로젝트를 만듭니다.</p>	
5	<p>빌드 타겟 설정을 필요한 대로 설정하여 응용 프로그램의 이미지를 빌드합니다. 7단계로 진행합니다.</p>	
6	<p>기존 CodeWarrior for RVDS 프로젝트를 엽니다.</p>	
7	<p>CodeWarrior for RVDS 프로젝트의 이미지를 빌드합니다.</p>	

표 기본 빌드 및 디버깅 작업 (계속)

단계	설명	참조
8	<p>사용할 이미지를 결정합니다.</p> <ul style="list-style-type: none"> • 미리 빌드된 예제 이미지와 같은 기존 이미지를 사용하려면 9단계로 진행합니다. • 새 이미지를 빌드하려면 1단계로 돌아갑니다. 	3-5페이지의 <i>예제 프로젝트 사용</i>
9	RealView Debugger를 시작합니다.	<i>RealView Debugger Essentials Guide</i>
10	디버그 타겟과 연결을 필요한 대로 구성합니다.	<i>RealView Debugger Target Configuration Guide</i>
	디버그 타겟에 연결합니다	<i>RealView Debugger Essentials Guide</i> <i>RealView Debugger User Guide</i>
	디버깅할 이미지를 로드합니다	<i>RealView Debugger Essentials Guide</i> <i>RealView Debugger User Guide</i>
	브레이크포인트 트레이스포인트 등의 디버깅 기능을 준비합니다	<i>RealView Debugger Essentials Guide</i> <i>RealView Debugger User Guide</i> <i>RealView Debugger Trace User Guide</i> <i>RealView Debugger RTOS Guide</i>
	이미지를 실행합니다	<i>RealView Debugger Essentials Guide</i> <i>RealView Debugger User Guide</i>
	단계별 실행 변수 및 메모리 내용 표시 등의 필요한 디버깅 및 모니터링 작업을 수행합니다. 트레이스포인트를 사용하는 경우 <i>의 트레이스 분석 기능을 사용하여 트레이스 출력을 분석합니다</i>	<i>RealView Debugger Essentials Guide</i> <i>RealView Debugger User Guide</i> <i>RealView Debugger Trace User Guide</i> <i>RealView Debugger RTOS Guide</i>
	<p>디버깅 세션의 결과에 따라 다음을 수행합니다</p> <p>문제가 있으면 <i>단계로 진행합니다</i></p> <p>문제가 없으면 최종 릴리스용 이미지를 다시 빌드합니다</p>	<ul style="list-style-type: none"> • <i>RealView Development Suite CodeWarrior IDE Guide</i> <i>RealView Debugger Essentials Guide</i> <i>RealView Compilation Tools 핵심 설명서</i>
	소스 코드의 문제를 해결할 방법을 결정합니다	

단계	설명	참조
	를 사용합니다	
	선택한 다른 소스 편집기를 사용합니다	
	문제를 수정한 후에는 이미지를 다시 빌드하고 다시 로드한 다음 디버깅해야 합니다	
	를 사용하는 경우 단계로 돌아갑니다	
	를 직접 사용하는 경우 단계로 돌아갑니다	

3.2 예제 프로젝트 사용

사용자 고유의 응용 프로그램 이미지를 빌드 및 디버깅하려는 경우에도 [설명서](#)에 설명된 작업에서는 [예제 프로젝트](#)와 함께 제공되는 예제 프로젝트 중 일부를 사용합니다. 자세한 내용은 [페이지](#)의 [예제](#)를 참조하십시오.

구성요소의 기능에 익숙해질 때까지는 설명된 지시 사항을 따르는 것이 좋습니다. 그러나 사용자 설명서에 설명된 대부분의 작업에서는 예제 파일을 수정해야 합니다. 예제 파일을 수정하려면 먼저 예제 프로젝트 파일과 디렉토리의 백업 복사본을 만들어야 합니다.

시작하기



부록 A

armenv 도구 사용

이 부록에서는 [armenv](#) 제품 설치를 관리하는 데 사용할 수 있는 **armenv** 도구에 대해 설명합니다. 이 부록에는 다음 단원이 포함되어 있습니다.

[페이지의 도구](#)

[페이지의 도구 사용](#)

도구

`armenv` 도구를 사용하면 다음을 수행할 수 있습니다.

- ARM RealView 제품의 환경 변수를 설정하고 제거합니다.
- 설치한 ARM RealView 제품 간의 충돌을 확인합니다.
- 동일한 제품의 여러 가지 변형을 설정합니다.

———— 참고 —————

이번 릴리스의 RVDS에서는 사용자 지정 설치에 `armenv` 도구를 사용할 수 없습니다.

`armenv` 도구는 다음 위치에 있습니다.

`install_directory/bin/platform`

A.2 armenv 도구 사용

이 단원에서는 `armenv` 명령 구문을 설명하고 몇 가지 예를 통해 명령 사용 방법을 보여 줍니다 이 단원에는 다음과 같은 내용이 포함되어 있습니다

명령 구문

명령 행 인수

페이지의 예

A.2.1 armenv 명령 구문

`armenv` 도구의 명령 구문은 다음과 같습니다

```
armenv [-r root] [-u] -p product [--and] -p product... [--user|--sys|--proc]
[--bat|--sh|--csh|--posh|--exec program [args]]
```

인수는 `명령 행 인수`에서 설명합니다

명령 행 인수

표 A-1에서는 모든 플랫폼에서 사용할 수 있는 명령 행 인수를 보여 줍니다

표 A-1 일반 `armenv` 인수

인수	설명
<code>--help</code>	명령 행 인수에 대한 도움말을 표시합니다
<code>-r root</code>	제품 설치 루트인 <code>install_directory</code> 의 절대 경로입니다 예를 들어 <code>C:\Program Files\ARM</code> 에서 기본 루트는 다음과 같습니다
<code>-p product</code>	제품입니다 자세한 내용은 <code>페이지의 제품 구문</code> 을 참조하십시오
<code>--and</code>	이 인수 앞에 있는 모든 제품의 설정을 처리한 다음 뒤에 있는 설정을 처리합니다 두 번째 그룹의 설정은 첫 번째 그룹의 설정보다 우선합니다

표 일반 인수 계속

인수	설명
--proc	현재 프로세스의 환경만 변경합니다 에서는 이 인수를 --system 또는 --user와 함께 사용할 수 없습니다
--exec	프로그램을 새 환경에서 실행합니다 이 인수는 에서 --bat와 함께 사용할 수 없으며 선 솔라리스 및 레드햇 리눅스에서는 --sh --csh 또는 --posh와 함께 사용할 수 없습니다
-u	제목을 설정할 때 환경에 대해 변경한 내용을 취소합니다

표 에서는 시스템에서 사용할 수 있는 명령 행 인수를 보여 줍니다

표 에서 사용할 수 있는 인수

인수	설명
--system	레지스트리의 Windows SYSTEM 영역을 업데이트합니다. 이 인수는 기본값입니다.
--user	레지스트리의 Windows USER 영역을 업데이트합니다.
--bat	현재 명령 프롬프트 창의 환경을 변경합니다. 이 인수는 기본값입니다.

표 에서는 선 솔라리스 및 레드햇 리눅스 시스템에서 사용할 수 있는 명령 행 인수를 보여 줍니다 이러한 인수 중 하나만 지정할 수 있습니다

표 선 솔라리스 및 레드햇 리눅스에서 사용할 수 있는 인수

인수	설명
--csh	csh 구문 셸 스크립트를 생성합니다.
--sh	sh 구문 셸 스크립트를 생성합니다.
--posh	이식 가능한 셸 스크립트를 생성합니다. 이 인수는 기본값입니다.

제품 구문

제품을 지정하기 위한 구문은 다음과 같습니다

```
-p category [name] [version [rev]] [-v name value]...
```

다음은 각 요소에 대한 설명입니다

<i>category</i>	제품 식별자 예 RVDS 또는 RVCT 입니다
<i>name</i>	이 인수는 사용하지 마십시오 armenv에서는 기본 이름인 Contents가 사용됩니다
<i>version</i>	제품의 버전 번호 예 3.0 입니다 버전을 지정하지 않으면 설치된 제품의 최신 버전이 사용됩니다
<i>rev</i>	제품의 특정 빌드 번호입니다 빌드 번호를 지정하지 않으면 설치된 제품의 최신 빌드가 사용됩니다

-v name value

동일한 제품의 변형을 식별합니다

name 변형 유형 예 platform 입니다 platform 변형만 사용하는 것이 좋습니다

value 특수 변형 예 solaris-sparc 입니다

예를 들어 버전 의 선 솔라리스 및 레드햇 리눅스 변형이 모두 설치되어 있을 수 있습니다 이 인수의 사용 방법을 보여 주는 예는 *예*를 참조하십시오

예

결과 버전 의 최신 빌드에 대해 선 솔라리스 환경 변수를 설정하려면 다음을 입력합니다

```
armenv -r ~/ -p RVDS 3.0 -v platform solaris-sparc --csh
```

버전 과 버전 간의 충돌을 확인하려면 다음을 입력합니다

```
armenv -p RVCT 3.0 -p RVCT 2.2
```

버전 설정이 버전 설정보다 우선하도록 하려면 다음을 입력합니다

```
armenv -p RVCT 2.2 --and -p RVCT 2.0
```


부록 B

이전 릴리스 정보

이 장에서는 *ARM* [®] *Developer Suite* 의 이전 릴리스인 버전 버전
버전 버전 과 *ARM Developer Suite*™ 버전 의 주요 차이점을 간략히
설명합니다. 다음 단원에서 변경 사항을 설명합니다.

 페이지의 *RVDS 버전 2.2* 에서 *RVDS 버전 2.2 SP1* 로 업그레이드되면서
 변경된 사항

 페이지의 *RVDS 버전 2.1* 에서 *RVDS 버전 2.2* 로 업그레이드되면서 변경
 된 사항

 페이지의 *RVDS 버전 2.0* 에서 *RVDS 버전 2.1* 로 업그레이드되면서 변경
 된 사항

 페이지의 *ADS 버전 1.2.1* 에서 *RVDS 버전 2.2* 로 업그레이드되면서 변경
 된 사항

B.1 RVDS 버전 2.2에서 RVDS 버전 2.2 SP1로 업그레이드되면서 변경된 사항

이 단원에서는 버전 2.2에서 버전 2.2 SP1로 업그레이드되면서 변경된 사항에 대해 설명합니다. 이 단원에는 다음 내용이 포함되어 있습니다.

설명서 변경 사항

디버거 지원

빌드 도구 지원

B.1.1 설명서 변경 사항

설명서의 변경 사항은 다음과 같습니다.

의 [*기능을 사용하는 방법을 설명하는](#)가 포함되었습니다.

- CodeWarrior 시작하기에 대해 설명된 장이 [시작 설명서](#)에서 제거되고 [시작 설명서](#)에 통합되었습니다.
- RealView Debugger 설명서에서 지원되는 DSP에 대한 내용이 변경되었습니다.

디버거 지원

RVDS 버전 2.2에서 RVDS 버전 2.2 SP1로 업그레이드되면서 디버깅 도구가 상당히 변경되어 RealView Debugger에서 CEVA-Oak, CEVA-TeakLite, CEVA-Teak, ZSP400 및 ZSP500 DSP를 지원하게 되었습니다.

빌드 도구 지원

RVDS 버전 2.2에서 RVDS 버전 2.2 SP1로 업그레이드되면서 빌드 도구가 약간 변경되었습니다. 자세한 내용은 [핵심 설명서](#)를 참조하십시오.

버전 2.1에서 버전 2.2로 업그레이드되면서 변경된 사항

이 단원에서는 RVDS 버전 2.1에서 RVDS 버전 2.2로 업그레이드되면서 변경된 사항에 대해 설명합니다. 이 단원에는 다음 단원이 포함되어 있습니다.

- [지원](#)
- [디버거 도구 지원](#)
- [B-4페이지의 빌드 도구 지원](#)
- [B-5페이지의 `gcc` 지원](#)

지원

RealView Debugger IDE를 대체하기 위해 CodeWarrior IDE가 제공되었습니다. RVDS 버전 2.2의 CodeWarrior IDE는 Metrowerks CodeWarrior 버전 5.6을 기반으로 합니다.

————— [참고](#) —————

RVDS 버전 2.2에서 CodeWarrior for RVDS는 Windows XP 및 Windows 2000 시스템에서만 지원되며 선 솔라리스 및 레드햇 리눅스용은 제공되지 않습니다.

CodeWarrior for RVDS에 대한 자세한 내용은 1-3페이지의 [코드워리어 IDE를 참조하십시오](#).

————— [참고](#) —————

CodeWarrior for ADS 사용자는 B-8페이지의 [코드워리어 IDE를 참조하십시오](#)에서 변경 사항을 참조하십시오.

디버거 도구 지원

RVDS 버전 2.1에서 RVDS 버전 2.2로 업그레이드되면서 디버깅 도구가 상당히 변경되어 RealView Debugger에 다음 기능이 포함되었습니다.

- 개선된 메뉴 구조
 - 개선된 창 처리 메커니즘
 - 새 Data Navigator 창을 사용한 개선된 데이터 탐색
 - 국제화 지원
 - 개선된 소스 코드 색 지정
 - 트레이스, 분석 및 프로파일링 향상
 - 향상된 RTOS 지원
 - `gcc`로 만든 이미지 지원
-

- 추가 CLI 명령인 PRINTDSM 및 TRACEEXTCOND

또한 독립 실행형 편집기와 Vi 편집 모드에 대한 지원이 RealView Debugger에서 제거되었습니다.

자세한 변경 사항 목록은 [이 페이지](#)를 참조하십시오.

빌드 도구 지원

RVDS 버전 2.1에서 RVDS 버전 2.2로 업그레이드되면서 빌드 도구가 다음과 같이 크게 변경되었습니다.

- RVCT 버전 2.2에는 ARM TrustZone® 기술에 최적화된 소프트웨어가 통합된 ARM1176JZF-S™, ARM968EJ-S, ARM1156T2F-S™, ARM MPCore™ 등의 새로운 ARMv6 코어에 대한 지원이 포함되었습니다.
- RVCT 버전 2.2에서 사용할 수 있는 새로운 Thumb®-2 명령어 세트에는 다양한 32비트 명령어와 몇 가지 16비트 명령어가 새로 도입되었습니다.
Thumb-2 명령어 세트에는 이전의 16비트 Thumb 명령어가 하위 세트로 포함되어 있습니다.
- RVCT 버전 2.2는 발표되지 않은 초안 버전의 [BPABI]와 완전하게 호환됩니다.
- RVCT 버전 2.2에서는 [이 페이지](#) 기본 표준[BSABI]에 기술된 대로 DWARF3(초안 표준 9) 디버그 테이블에 대한 초기 지원을 제공합니다.
- 명령 행 옵션 -g는 현재 컴파일에 대한 디버그 테이블 생성 기능을 설정합니다. 최적화 옵션은 -O로 지정됩니다. 기본적으로 -g 옵션을 사용해도 최적화 설정에는 영향을 주지 않습니다.
이는 RVCT 버전 2.2에서 변경된 동작입니다.
- RVCT 버전 2.2에서는 System V 공유 라이브러리와 호환되는 코드를 컴파일하는 명령 행 옵션 --apcs /fpic를 지원합니다.
- ARM 링커에서는 공유 라이브러리에 대한 빌드 및 연결을 지원합니다. 새 명령 행 옵션을 사용하여 SVr4 실행 가능 파일과 공유 개체를 빌드하고 코드가 생성되는 방식을 지정할 수 있습니다.
- ARM 링커에서는 GNU 확장 심볼 버전 관리 모델을 지원합니다.

- 부동 소수점 처리의 ARM 구현이 변경되어 C99 함수에 대한 개선된 지원 기능을 제공합니다. 이로 인해 동작이 상당히 변경된 부분에서는 개발자가 코드를 마이그레이션하여 새 기능을 사용할 수 있도록 호환 모드가 도입되었습니다.
- RVCT 버전 2.2에서는 리눅스 응용 프로그램 및 공유 라이브러리의 빌드를 지원합니다.

자세한 변경 사항 목록은 *RealView Compilation Tools 핵심 설명서*를 참조하십시오.

지원

Agilent Probe 지원은 현재 RVDS 버전 2.2의 사용자 지정 설치 옵션으로 사용할 수 있습니다.



B.3 RVDS 버전 2.0에서 RVDS 버전 2.1로 업그레이드되면서 변경된 사항

이 단원에서는 버전 2.0에서 버전 2.1로 업그레이드되면서 변경된 사항에 대해 설명합니다 이 단원에는 다음 단원이 포함되어 있습니다

디버거 도구 지원

빌드 도구 지원

B.3.1 디버거 도구 지원

버전 2.0에서 버전 2.1로 업그레이드되면서 디버깅 도구가 다음과 같이 크게 변경되었습니다

가 포함되어 있습니다 자세한 내용은 페이지의
를 참조하십시오

의
가 포함되어 있습니다 자세한 내용은 페이지
를 참조하십시오

에 다음 기능이 포함되었습니다

트레이스 및 프로파일링 향상

향상된 지원

일반적으로 사용하는 기능에 빠르게 액세스할 수 있도록 도구 모음 단추가 새로 추가되고 메뉴가 변경되었습니다

B.3.2 빌드 도구 지원

버전 2.0에서 버전 2.1로 업그레이드되면서 빌드 도구가 다음과 같이 크게 변경되었습니다

기본 표준 즉

기본 표준과의 호환성이 향상되었습니다 자세한 내용은
<http://www.arm.com/>에서 페이지를 참조하십시오

예외 처리가 지원됩니다 따라서 *ISO/IEC 14822 :1998 International Standard for C++*에 정의된 대로 내보내기 템플릿의 예외와 함께 의 나머지가 지원됩니다

다중 파일 컴파일 및 링커 피드백 등의 추가 최적화 기능이 포함되었습니다

이미지 크기를 크게 줄일 수 있는 읽기 쓰기 데이터 영역의 압축 기능이 제공됩니다

일부 및 확장이 지원됩니다

빌드 도구에 새 명령 행 옵션이 여러 가지 추가되었습니다

단일 대시 키워드와 일부 명령 행 옵션은 향후 제공되지 않을 예정입니다

————— 참고 —————

도구는 바이트 정렬을 유지하는 데 보다 엄격해졌습니다. 컴파일러는 **PRESERVE8** 및 **REQUIRE8**로 코드를 생성하고 링커는 바이트 정렬을 필요로 하는 코드가 바이트 정렬을 유지하는 코드만 호출하는지 확인합니다. 따라서 레거시 어셈블러 코드 개체 파일 및 라이브러리와 밀접한 관련이 있습니다. 기존 어셈블리 파일 개체 파일 또는 라이브러리가 바이트 정렬을 유지하는지 확인하고 필요한 경우 이를 수정해야 합니다. 자세한 내용은 *RealView Compilation Tools 어셈블러 설명서* 및 *RealView Compilation Tools 링커 및 유틸리티 설명서*를 참조하십시오.

B.4 ADS 버전 1.2.1에서 RVDS 버전 2.2로 업그레이드되면서 변경된 사항

이 단원에서는 버전 1.2.1에서 버전 2.2로 업그레이드되면서 변경된 사항에 대해 설명합니다 이 단원에는 다음 단원이 포함되어 있습니다

변경 사항

페이지의 디버거 변경 사항

페이지의 빌드 도구 변경 사항

페이지의 시뮬레이터 변경 사항

B.4.1 CodeWarrior IDE 변경 사항

이 단원에서는 버전 1.2.1에서 버전 2.2로 업그레이드되면서 다음 사항이 변경되었습니다

이 단원에서는 버전 1.2.1을 기반으로 하지만 버전 2.2를 기반으로 합니다

버전 1.2.1의 스크립트를 처리하기 위한 지원 기능을 제공하는 플러그인인 이 버전에서는 제거되었습니다 따라서 이 플러그인은 더 이상 버전 2.2에서 지원하지 않습니다

도구 관련 구성 패널이 버전 2.2에 맞게 조정되었습니다

버전 1.2.1에서는 개별 컴파일러가 단일 컴파일러로 결합되었으므로 버전 2.2에는 컴파일러 구성 패널이 하나만 있습니다

및 외에도 를 사용하여 이미지를 실행하고 디버깅할 수 있게 되었습니다

여러 개의 라이브러리를 연결할 수 있습니다

프로젝트를 로 가져올 수 있습니다

의 기본 샘플에는 DebugRe1 빌드 타겟이 포함되어 있지 않습니다 그러나 프로젝트를 가져오는 경우에는 DebugRe1 빌드 타겟이 만들어지므로 해당 빌드 타겟에 대해 구성된 설정을 유지할 수 있습니다

컴파일러는 컴파일러와 달리 브라우저 정보를 생성하지 않습니다 이 기능은 현재 의 기본 제공 언어 파서를 통해 제공됩니다

코드 서식

- 템플릿 클래스의 코드 완성을 비롯한 코드 완성 기능
- 다음 이전 함수로 이동 기능
- 인쇄 시 줄 바꿈 기능
- 소스 관련 `#includes` 지원
- 주석 안쪽 바깥쪽 찾기 기능
- 개선된 언어 파서 속도 및 피드백
- 새 편집기 바인딩
- 프로젝트 창에 및 열을 표시하거나 숨길 수 있는 기능
- 작업 공간 지원

참고

의 타겟 연결 및 디버깅 기능은 모두 에서 사용할 수 없습니다 디버거 중 하나를 실행하여 해당 기능을 수행해야 합니다

B.4.2 디버거 변경 사항

버전 에서 버전 로 업그레이드되면서 디버깅 도구가 다음과 같이 크게 변경되었습니다

는 다음과 같은 고급 디버깅 기능을 수행할 수 있는 최신 디버거입니다

- 다중 프로세서 디버깅
- 인식 디버깅
- 확장 타겟 가시성
- 트레이스 분석 및 프로파일링
- 이더넷 및 를 통한 제어 장치 액세스

는 버전 와 함께 제공되는 새로운 를 사용하여 빌드된 및 프로그램을 디버깅할 수 있도록 향상되었습니다

빌드 도구 변경 사항

버전 1.10.0 에서 버전 1.11.0 로 업그레이드되면서 빌드 도구가 다음과 같이 크게 변경되었습니다

새로운 `arm` 기본 표준을 준수합니다. 자세한 내용은 <http://www.arm.com/>에서 페이지를 참조하십시오. 이는 이전 버전에서 달라진 점입니다. 일부 호환성은 `--apcs /adsabi` 명령 행 옵션을 통해 제공됩니다.

이전에 정의된 모든 `__asm__` 지원이 프런트 엔드를 통해 제공됩니다. 여기에는 예외 네임스페이스 템플릿 및 `__asm__` 형식 정보의 자동 구현이 포함되지만 템플릿 내보내기는 제외됩니다.

일부 언어 확장에 대한 지원이 포함되었습니다.

함수 단위의 `__asm__` 및 `__asm__` 컴파일이 가능합니다.

인라인 어셈블러가 다시 엔지니어링되고 라인 외부 어셈블리 코드를 포함할 수 있게 해 주는 새로운 임베디드 어셈블러가 추가되었습니다.

사용하지 않는 함수를 제거하기 위한 링커 피드백이 포함되었습니다.

`__asm__` 아키텍처 버전 명령어 전체에 대한 지원이 추가되었습니다.

읽기 쓰기 데이터 압축을 통해 `__asm__` 크기를 최적화할 수 있습니다.

사용하지 않는 가상 함수가 제거되었습니다.

여러 컴파일 단위에서 최적화를 수행하는 다중 파일 컴파일 기능이 포함되었습니다.

라이브러리 검색 경로를 지정하여 사용자 라이브러리를 검색할 위치를 설정할 수 있습니다.

코드와 데이터를 서로 다른 실행 영역에 분리할 수 있습니다.

새로운 스캐터 로드 특성이 포함되었습니다.

유니코드 및 다중 바이트 문자가 지원됩니다.

함수의 반환 주소, 현재 스택 포인터 값 및 현재 프로그램 카운터 값에 액세스하는 데 컴파일러 `__asm__` 함수를 사용할 수 있습니다. 추가 `__asm__` 함수를 사용하면 `__asm__` 코드에 `BKPT` 명령어를 삽입할 수 있습니다.

반환되지 않는 함수를 식별할 수 있으므로 컴파일러에서 보다 효율적인 코드를 생성합니다

이름 변환 체계가 변경되었습니다

에서는 *프로파일러 armprof* 가 제공되지 않습니다

에서는 가 제공되지 않습니다

컴파일러는 컴파일러와 달리 브라우저 정보를 생성하지 않습니다

어셈블러 컴파일러 및 링커 명령 행 옵션이 변경되었습니다

이중 대시 로 명령 행 키워드를 나타내고 예 `--cpp` 단일 대시 로 인수가 포함되거나 포함되지 않는 명령 행 단일 문자 옵션을 나타낼 수 있습니다 예 `-S`

————— 참고 —————

이전 버전의 및 에서 사용되는 단일 대시 명령 행 옵션은 역방향 호환성을 위해 여전히 지원됩니다

옵션인 `-ihf`가 제거되었습니다

————— 참고 —————

도구는 바이트 정렬을 유지하는 데 보다 엄격해졌습니다 컴파일러는 `PRESERVE8` 및 `REQUIRE8`로 코드를 생성하고 링커는 바이트 정렬을 필요로 하는 코드가 바이트 정렬을 유지하는 코드만 호출하는지 확인합니다 따라서 레거시 어셈블러 코드 개체 파일 및 라이브러리와 밀접한 관련이 있습니다 기존 어셈블러 파일 개체 파일 또는 라이브러리가 바이트 정렬을 유지하는지 확인하고 필요한 경우 이를 수정해야 합니다 자세한 내용은 *어셈블러 설명서* 및 *링커 및 유틸리티 설명서*를 참조하십시오

시뮬레이터 변경 사항

는 최신 버전의 시뮬레이터로
및 를 통한 연결을 지원합니다 에서
를 통해 시뮬레이터에 연결할 경우 해당 시뮬레이터에 대한 연결을 여러 개 설정할 수 있습니다 버전 및 을 사용하여
의 인터페이스에 연결할 수 있습니다

참고

버전 뿐만 아니라 도 설치할 수 있지만
와 를 모두 사용하는 경우에는 매우 조심해야 합니다
자세한 내용은 *버전 릴리스 정보*를 참조하십시오
