

DVEVM

入門マニュアル

DVEVM

入門マニュアル

対応英文マニュアル：SPRUE66B
2007 年 3 月

2007 年 4 月

この資料は、Texas Instruments Incorporated (TI) が英文で記述した資料を、皆様のご理解の一助として頂くために日本テキサス・インスツルメンツ (日本 TI) が英文から和文へ翻訳して作成したものです。資料によっては正規英語版資料の更新に対応していないものがあります。日本 TI による和文資料は、あくまでも TI 正規英語版をご理解頂くための補助的参考資料としてご使用下さい。製品のご検討およびご採用にあたりましては必ず正規英語版の最新資料をご確認下さい。

TI および日本 TI は、正規英語版にて更新の情報を提供しているにもかかわらず、更新以前の情報に基づいて発生した問題や障害等につきましては如何なる責任も負いません。



最新の英語版資料

<http://focus.ti.com/lit/ug/sprue66b/sprue66b.pdf>

ご注意

日本テキサス・インスツルメンツ株式会社（以下TIJといいます）及びTexas Instruments Incorporated (TIJの親会社、以下TIJおよびTexas Instruments Incorporatedを総称してTIといいます）は、その製品及びサービスを任意に修正し、改善、改良、その他の変更をし、もしくは製品の製造中止またはサービスの提供を中止する権利を留保します。従いまして、お客様は、発注される前に、関連する最新の情報を取得して頂き、その情報が現在有効かつ完全なものであるかどうかご確認下さい。全ての製品は、お客様とTIとの間に取引契約が締結されている場合は、当該契約条件に基づき、また当該取引契約が締結されていない場合は、ご注文の受諾の際に提示されるTIの標準契約約款に従って販売されます。

TIは、そのハードウェア製品が、TIの標準保証条件に従い販売時の仕様に対応した性能を有していること、またはお客様とTIとの間で合意された保証条件に従い合意された仕様に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する固有の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

TIは、製品のアプリケーションに関する支援もしくはお客様の製品の設計について責任を負うことはありません。TI製部品を使用しているお客様の製品及びそのアプリケーションについての責任はお客様にあります。TI製部品を使用したお客様の製品及びアプリケーションについて想定される危険を最小のものとするため、適切な設計上および操作上の安全対策は、必ずお客様にてお取り下さい。

TIは、TIの製品もしくはサービスが使用されている組み合わせ、機械装置、もしくは方法に関連しているTIの特許権、著作権、回路配置利用権、その他のTIの知的財産権に基づいて何らかのライセンスを許諾することは明示的にも黙示的にも保証も表明もしておりません。TIが第三者の製品もしくはサービスについて情報を提供することは、TIが当該製品もしくはサービスを使用することについてライセンスを与えるとか、保証もしくは是認することなどを意味しません。そのような情報を使用するには第三者の特許その他の知的財産権に基づき当該第三者からライセンスを得なければならない場合もあり、またTIの特許その他の知的財産権に基づきTIからライセンスを得て頂かなければならない場合もあります。

TIのデータ・ブックもしくはデータ・シートの中にある情報を複製することは、その情報に一切の変更を加えること無く、且つその情報と結び付けられた全ての保証、条件、制限及び通知と共に複製がなされる限りにおいて許されるものとします。当該情報に変更を加えて複製することは不正で誤認を生じさせる行為です。TIは、そのような変更された情報や複製については何の義務も責任も負いません。

TIの製品もしくはサービスについてTIにより示された数値、特性、条件その他のパラメーターと異なる、あるいは、それを超えてなされた説明で当該TI製品もしくはサービスを再販売することは、当該TI製品もしくはサービスに対する全ての明示的保証、及び何らかの黙示的保証を無効にし、且つ不正で誤認を生じさせる行為です。TIは、そのような説明については何の義務も責任もありません。

なお、日本テキサス・インスツルメンツ株式会社半導体集積回路製品販売用標準契約約款もご覧下さい。

<http://www.tij.co.jp/jsc/docs/stdterms.htm>

Copyright © 2007, Texas Instruments Incorporated

日本語版 日本テキサス・インスツルメンツ株式会社

弊社半導体製品の取り扱い・保管について

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様の実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

1. 静電気

- 素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。
- 弊社出荷梱包単位(外装から取り出された内装及び個装)又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で(導電性マットにアースをとったもの等)、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使うこと。
- マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気の帯電を防止する措置を施すこと。
- 前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。

2. 温・湿度環境

- 温度：0～40℃、相対湿度：40～85%で保管・輸送及び取り扱いを行うこと。(但し、結露しないこと。)

- 直射日光があたる状態で保管・輸送しないこと。
3. 防湿梱包
 - 防湿梱包品は、開封後は個別推奨保管環境及び期間に従い基板実装すること。
 4. 機械的衝撃
 - 梱包品(外装、内装、個装)及び製品単品を落下させたり、衝撃を与えないこと。
 5. 熱衝撃
 - はんだ付け時は、最低限260℃以上の高温状態に、10秒以上さらさないこと。(個別推奨条件がある時はそれに従うこと。)
 6. 汚染
 - はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質(硫黄、塩素等ハロゲン)のある環境で保管・輸送しないこと。
 - はんだ付け後は十分にフラックスの洗浄を行うこと。(不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。)

以上

評価用ボード、キットに関する重要なお知らせ

テキサス・インスツルメンツ（以下、TI と言います）は、同梱された製品を以下の条件で提供いたします。

本評価用ボードないしキットは、技術開発、デモンストレーション、若しくは評価目的にのみ使用されると想定されています。従って、TI は、本品が一般的消費者のための完成品であるとは見做していません。本品を取り扱う方は、電子工学に関する実務経験を有し、且つ良識ある技術的実務基準に従って取り扱う方でない限りはなりません。それゆえ、本品は、半導体集積回路製品や回路基盤を組み込んだ最終製品において通常要求されるような製造物安全や環境保全のための手段を含む設計上、販売上、若しくは製造上の保護的措置に関しては、未完成品であります。

本評価用ボードないしキットには、電磁気適合性に関する EU の指令、制限下にある物質（RoHS）、リサイクル（WEEE）、FCC、CE、UL 等に関する基準は適用されませんので、従って、これらの指令若しくはその他の指令の技術的要求事項には適合しない可能性があります。

もし、本評価用ボードないしキットがユーザーズ・ガイドに示された仕様に合致しない場合は、本品の送付から 30 日以内に返品して頂ければ本品に対して支払われた代金の全額を返金いたします。**本保証は、TI が本品のお客様に対して提供する唯一の保証であり、商品性があることの保証、特定目的に合致することの保証を含めた明示的保証、黙示的保証、法定の保証その他ありとあらゆる保証を排除して適用される保証であります。**

本品の使用人は、本品を適正且つ安全に使用することについての全責任を負うものとします。さらに、もし万一使用者による本品の取扱いにより TI が何らかの請求、訴え等を提起された場合は、TI に補償を提供するものとします。本品は、開放的構造になっているため、使用者は、その責任をもって、静電気放電（ESD）に関する適切な予防対策で必要なものを全てを取らなくてはならないものとします。

上記に規定された補償を除き、いずれの当事者も、間接的、特別的、偶発的、派生的損害については責任を負わないものとします。

TI は、現在、多数の顧客と本品に関して取引を行っているため、TI とお客様（貴社、貴殿）との本品に関する取引は、排他的なものではないものとします。

TI は、本品を利用するお客様の製品に関する支援、お客様の製品設計、ソフトウェアが動くかどうか、特許侵害、もしくはここに記載されている役務の提供については一切責任を負わないものとします。

本品を取り扱う前に、必ず、ユーザーズ・ガイドをご覧ください。とりわけ、ユーザーズ・ガイドの中の「警告と禁止事項（Warning and Restrictions）」に関するお知らせをご覧ください。そのお知らせには、温度と電圧に関する安全についての重要な情報が含まれています。TI の環境ないし安全に関するプログラムについての追加的情報を得るためには、TI のアプリケーション・エンジニアに連絡して頂くか、若しくは TI のウェブ・サイト www.tij.co.jp/jcorp/docs/esh をご覧下さい。

TI は、本品の提供によって、本品若しくは役務が使用され得る若しくは使用されているところの機械、方法、組み合わせをカバーする若しくはそれらに関する特許、その他の知的財産権を許諾するものではないものとします。

FCCに関する警告

本評価ボードないしキットは、技術開発、デモンストレーション、若しくは評価目的にのみ使用されると想定されており、従って TI は、本品が一般的消費者のための完成品であるとは見做していません。本品は、高周波（RF）エネルギーを発生、使用、かつ放射し得ることがあり、且つ電波干渉に対抗するための適切な保護を提供する目的で設定された FCC 規則第 15 章に従ったコンピューティング・デバイスの制限に適合するか否かの試験は行われておりません。本装置を、無線通信に対する干渉が起り得る他の環境下で操作する場合は、使用者は、自らの費用により、当該干渉を是正するために必要とされる何らかの手段を取らなくてはならないものとします。

まえがき

本書について

DVEVM（デジタルビデオ評価モジュール）は、DM644x アーキテクチャをご覧ください、マルチメディア・エンジンとしての DM644x の能力と性能をユーザに評価していただくための評価プラットフォームです。本書の想定読者は、DM644x ARM コア上で Linux ベースのソフトウェアの開発しているユーザです。

DVEVM では、ソフトウェア開発を行うための DSP コアを公開していません。DSP コアは、すぐにコーデックを実現するための「ブラックボックス」として扱っています。DVSDK でアップグレードすると、DSP 側のアプリケーション開発能力を追加することができます。さらに、DVSDK を使うと完全な Linux のライセンスが追加されます。

本書では、ボードに関する概要と付属ソフトウェアについて説明します。DVEVM について「理解を深める」最初の資料として、読者の方々がご使用になることを想定しています。他の資料は、さらに理解を深めるためにご利用ください。このような資料の完全なリストについては、この製品に同梱している DVEVM の資料目録を参照してください。

補足資料およびリソース

本書の参考資料を次に示します。

- ❑ Spectrum Digital 社の DaVinci EVM ホーム：
<http://c6000.spectrumdigital.com/davincievms/>
- ❑ DaVinci プロセッサ情報を共有する TI Linux コミュニティ：
<http://linux.davincidsps.com>
- ❑ 『Codec Engine Application Developer's Guide』（資料番号 SPRUE67）

表記規則

本書では、次の表記規則を使用します。

- ❑ プログラム・リスト、プログラム例、および対話表示は、タイプライタの活字に似た特殊な活字（mono-spaced font）で示してあります。例は、強調のため、ボールド（**bold**）で示してあります。対話表示についても、ユーザが入力するコマンドとシステムが表示する項目（プロンプト、コマンド出力、エラー・メッセージなど）と区別するために、ボールド（**bold**）で示しています。
- ❑ 大括弧（[および]）は、オプションのパラメータを識別します。パラメータを使用する場合、指定内容はこの括弧内に入力します。大括弧が**ボールド**でない限り、括弧そのものは入力する必要がありません。

商標

Texas Instruments ロゴおよび Texas Instruments は、Texas Instruments の登録商標です。TI、DaVinci、DaVinci ロゴ、XDS、Code Composer、Code Composer Studio、Probe Point、Code Explorer、DSP/BIOS、RTDX、Online DSP Lab、DaVinci、TMS320、TMS320C54x、TMS320C55x、TMS320C62x、TMS320C64x、TMS320C67x、TMS320C5000、および TMS320C6000 は、Texas Instruments の商標です。



MS-DOS、Windows、および Windows NT は、Microsoft Corporation の商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Linux は Linus Torvalds の登録商標です。

Solaris、SunOS、および Java は、Sun Microsystems, Inc. の商標または登録商標です。

その他のすべてのブランド名、製品名、およびサービス名は、それぞれの会社または団体の商標あるいは登録商標です。

注意

製品に同梱されているペリフェラルの型番や仕様は変更される場合があります。

目次

1	DVEVM の概要.....	1-1
	本章では、DVEVM（デジタルビデオ評価モジュール）の概要を説明します。	
1.1	はじめに.....	1-2
1.2	キットの内容.....	1-3
1.3	ボードの構成.....	1-4
1.4	次に行うこと.....	1-5
2	DVEVM ハードウェアのセットアップ.....	2-1
	本章では、DVEVM ハードウェアのセットアップ方法について説明します。	
2.1	ハードウェアのセットアップ方法.....	2-2
2.2	コンソール・ウィンドウに接続する方法.....	2-6
3	デモンストレーション・ソフトウェアの実行方法.....	3-1
	本章では、DVEVM に付属しているデモンストレーション・ソフトウェアの実行方法について説明します。	
3.1	デフォルトのブート構成.....	3-2
3.2	スタンドアロン・デモの開始方法.....	3-2
3.3	スタンドアロン・デモの実行方法.....	3-3
3.3.1	Encode + Decode デモについて.....	3-5
3.3.2	Encode デモについて.....	3-6
3.3.3	Decode デモについて.....	3-8
3.3.4	Third Party Menu について.....	3-9
3.4	ネットワーク・デモの実行方法.....	3-10
4	DVEVM ソフトウェアのセットアップ.....	4-1
	本章では、DVEVM に付属しているソフトウェアの使用方法について説明します。	
4.1	ソフトウェアの概要.....	4-2
4.1.1	本書のコマンド・プロンプト.....	4-3
4.1.2	ソフトウェア・コンポーネント.....	4-4
4.2	インストールの準備.....	4-5
4.3	ソフトウェアのインストール方法.....	4-6
4.3.1	ターゲット Linux ソフトウェアのインストール方法.....	4-6
4.3.2	DVEVM ソフトウェアのインストール方法.....	4-7
4.3.3	A/V デモ・ファイルのインストール方法.....	4-8
4.3.4	ターゲットからのアクセス用に共有ファイル・システムをエクスポートする 方法.....	4-8
4.3.5	共有ファイル・システムのテスト.....	4-9

4.3.6	PAL ビデオ・ユーザ用のブート・セットアップの構成.....	4-10
4.4	ビルド / 開発環境のセットアップ方法.....	4-11
4.4.1	簡単なプログラムを作成して DVEVM 上で実行する方法.....	4-11
4.5	新しい Linux カーネルのビルド方法	4-12
4.6	ターゲットに合わせて DVEVM ソフトウェアを再ビルドする方法	4-13
4.7	新しい Linux カーネルのブート方法	4-14
4.8	アップグレードのインストール	4-14
A	追加手順	A-1
	付録では、ユーザのセットアップ内容および固有のニーズに基づいて使用できるオプションの 手順について説明します。	
A.1	コマンドラインからデモを実行する方法.....	A-2
A.2	ビデオ入出力方式の変更方法.....	A-6
A.3	デモ・アプリケーションを Third Party Menu に追加する方法.....	A-9
A.4	TFTP サーバのセットアップ方法	A-11
A.5	その他のブート方法.....	A-12
A.6	Windows でのコンポーネントのインストール方法	A-16
A.7	DSP/BIOS Link の再ビルド方法.....	A-17
A.8	DVEVM ハードディスク・ドライブの復元および更新方法.....	A-18

DVEVM の概要

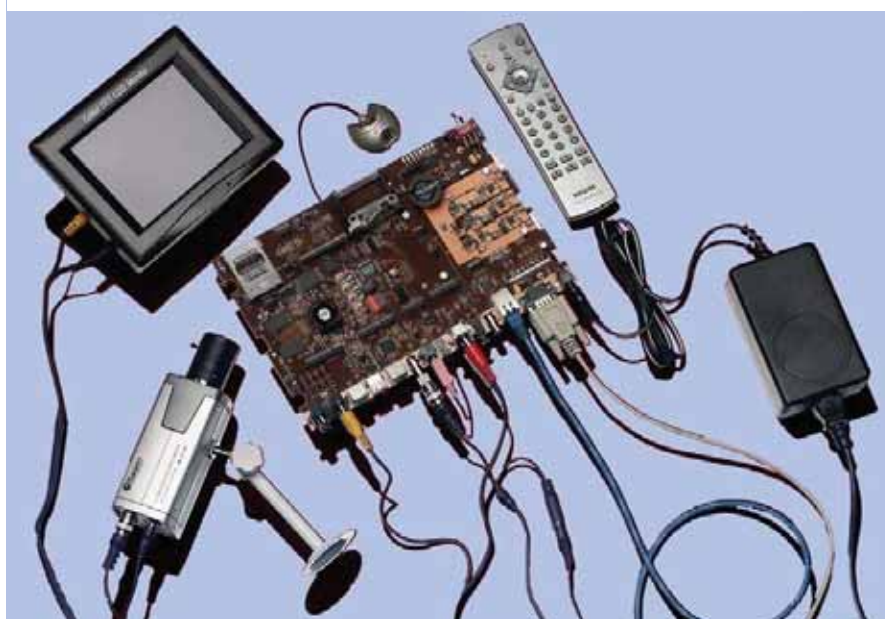
本章では、DVEVM (デジタルビデオ評価モジュール) の概要を説明します。

項目	ページ
1.1 はじめに	1-2
1.2 キットの内容	1-3
1.3 ボードの構成	1-4
1.4 次に行うこと	1-5

1.1 はじめに

新製品 DVEVM (デジタルビデオ評価モジュール) を使用すると、TI の新しい DaVinci™ テクノロジーおよび DM644x アーキテクチャを評価することができます。

DaVinci テクノロジーにより、効率性を考慮したシステムソリューション・コンポーネントとすばらしいデジタル・ビデオとオーディオが融合されました。



本書の対象読者は、DM644x ARM コア上で Linux ベースのソフトウェアの開発しているユーザを想定しています。DVEVM では、ソフトウェア開発を行うための DSP コアを公開していません。DSP コアは、すぐにコーデックを実現するための「ブラックボックス」として扱っています。Texas Instruments のアップグレード・キット (別売) を使用すると、DSP の開発を行うためのツールやソフトウェアが手に入ります。

1.2 キットの内容

DVEVM キットには、次のハードウェアが付属しています。これらのコンポーネントの接続方法については、2.1 節「ハードウェアのセットアップ方法」で説明します。

- ❑ **DVEVM ボード** (SDI P/N 702050)。この DVEVM ボードには、汎用プロセッサと高速化された DSP プロセッサの両方の機能を使うアプリケーション開発用に、ARM9 および C64+ DSP を組み込んだ DaVinci TMS320DM6446 デュアルコア・デバイスが搭載されています。
- ❑ **ハードディスク・ドライブ** (Samsung P/N MP0402H)。DVEVM には、40 GB の容量をもつ 2.5 インチの Spinpoint ハードディスク・ドライブが付属しています。このドライブの回転速度は 5400 RPM で、キャッシュは 8 MB です。このドライブは Ultra ATA 66/100/133 IDE に対応しています。DVEVM ボードのハードディスク・ドライブには、ソフトウェアが事前にインストールされています。
- ❑ **CCD カメラ** (Swann P/N SW-C-C500R)。この CCD カメラは、DaVinci アプリケーションで NTSC または PAL ビデオ画像処理機能を提供します。
- ❑ **LCD ディスプレイ** (Delvcam P/N DELVPRO56 (NTSC) / DELVPRO56PL (PAL))。DVEVM キットに付属している Delvcam LCD ディスプレイの画面サイズは、5.6 インチ (320x240 ピクセル) です。各種ケーブルおよび電源ケーブルが付属しています。NTSC バージョンは、110 VAC 電源です。PAL バージョンは、220 VAC 電源です。
- ❑ **PC デスクトップ・マイクロフォン** (Labtec Verse 333)。マイクロフォンは、DaVinci アプリケーションで利用するためにオーディオを取り込む方法を提供します。
- ❑ **赤外線リモート・コントロール** (Phillips Magnavox P/N PM4S EFM7ND)。このユニバーサル・リモート・コントロールは、アプリケーションのデモを行うためのユーザ・インターフェイスを提供するために組み込まれています。
- ❑ **A/V ケーブル**。DVEVM ボードを、開発を行うために使用するペリフェラル・デバイスやホストの Linux ワークステーションに接続するためのケーブルがキットに付属しています。

DVEVM キットには、次のソフトウェア CD-ROM も付属しています。ソフトウェア・コンポーネントの使用方法については、第 4 章を参照してください。

- ❑ DaVinci デジタルビデオ評価キット。
- ❑ MontaVista Linux Pro v4.0 ターゲットの TI DaVinci デモ・バージョン。
- ❑ MontaVista Linux Pro v4.0 ツールの TI DaVinci デモ・バージョン。
- ❑ 補足のソフトウェア CD-ROM は DVSDK アップグレード・キットで提供されます。

The logo for DVSDK is located in the bottom left corner of the page. It consists of the text "DVSDK" in a bold, sans-serif font, slanted upwards to the right, and enclosed within a dark, parallelogram-shaped border.

1.3 ボードの構成

DVEVM は、マルチメディア・アプリケーションで利用可能なペリフェラルが組み込まれた状態になっています。またボード上のハードディスク・ドライブには、デモンストレーション・ソフトウェアが事前にインストールされた状態にもなっています。次のブロック図に、主なハードウェア・コンポーネントを示します。

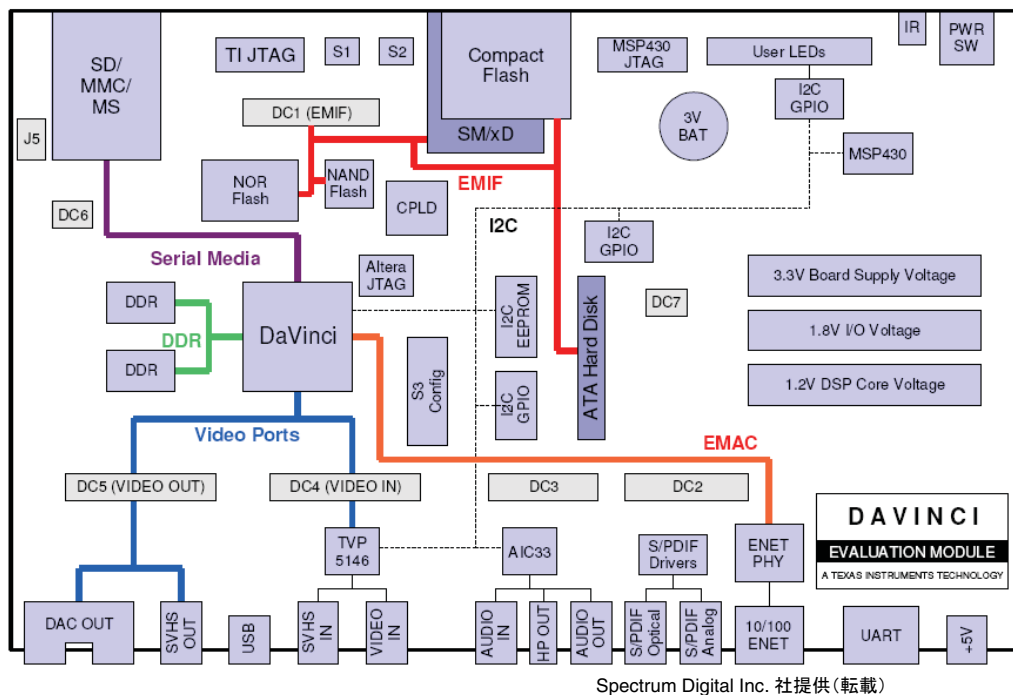


図 1-1. DVEVM ハードウェア・ブロック図

DVEVM ハードウェアの詳細は、DaVinci EVM の Web サイト (<http://c6000.spectrumdigital.com/davincievml/>) をご覧ください。

DaVinci EVM には、ボードに電力が供給されない時に MSP430 のリアルタイム・クロックにバックアップ電力を供給するための、バッテリー・ホルダーがあります。バッテリーはこのキットには付属していません。推奨バッテリーのパーツ番号については、Spectrum Digital 社の DaVinci EVM 技術資料を参照してください。

1.4 次に行うこと

DVEVM を評価し、DM644x 用のアプリケーションの開発を始めるためには、まず本書をお読みください。本書では、ハードウェアを接続することからはじめて、ソフトウェアをテストし、アプリケーションを開発するところまでを順に説明します。

DaVinci テクノロジーおよび DM644x アーキテクチャの詳細を理解している場合には、次の内容をご覧ください。

- ❑ Spectrum Digital 社の DaVinci EVM ホーム :
<http://c6000.spectrumdigital.com/davincievms/>
- ❑ DaVinci プロセッサ情報を共有する TI Linux コミュニティ :
<http://linux.davincidsps.com>
- ❑ 『Codec Engine Application Developer's Guide』(資料番号 SPRUE67)
- ❑ DVEVM に付属する CD-ROM に含まれる PDF 形式の他の資料

次に行うこと

DVEVM ハードウェアのセットアップ

本章では、DVEVM ハードウェアのセットアップ方法について説明します。

項目	ページ
2.1 ハードウェアのセットアップ方法.....	2-2
2.2 コンソール・ウィンドウに接続する方法.....	2-6

2.1 ハードウェアのセットアップ方法

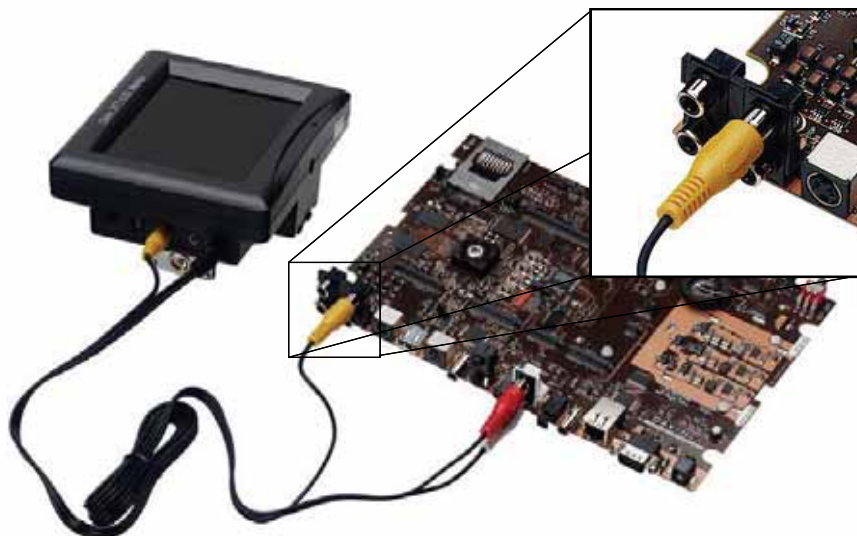
DVEVM に付属しているハードウェアをセットアップするには、これから説明する手順に従ってください。特定のペリフェラルにアクセスする必要がある場合は、説明を読まずに先へ進むことができます。たとえば、シリアルケーブルを使う必要がない場合には、先へ進んでください。

- 1) DVEVM は静電放電 (ESD) の影響を受けやすいので、ボードが損傷しないように、静電気防止用ストラップまたは他のデバイスを使用してください。

機器の電源を入れる前には必ず、通信ケーブルを接続してください。

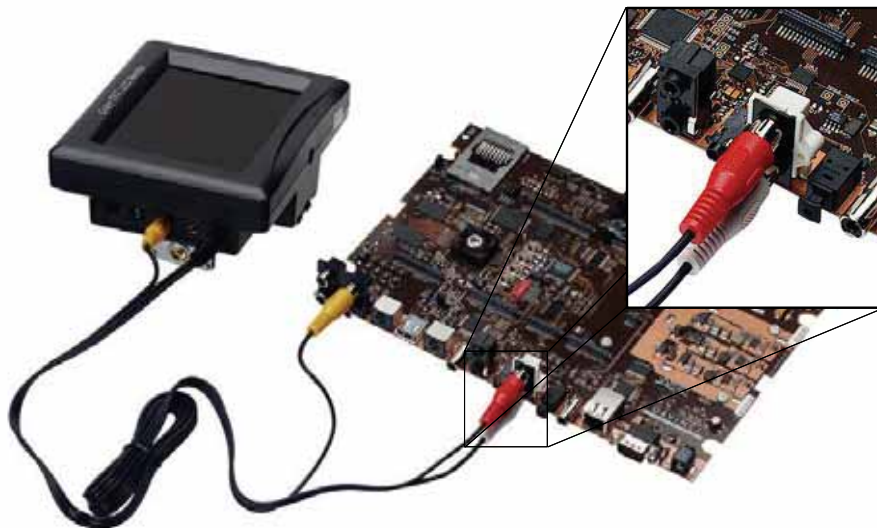


- 2) PAL ビデオを使用する場合には、S3 (USER) スイッチ・バンクのディップスイッチ 10 をオンにセットします。NTSC ビデオを使用する場合には、このディップスイッチ 10 をオフにセットします。S3 スイッチ・バンクの位置は、図 1-1 を参照してください。
- 3) 黄色のビデオ・ケーブルを DVEVM の右上のビデオ出力ジャックと接続し、LCD ディスプレイのビデオ入力に接続します (下記参照)。

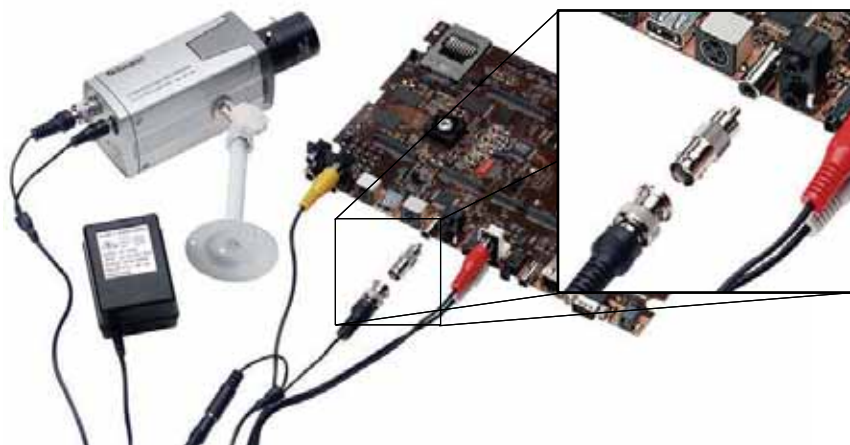


S ビデオまたはコンポーネント・ビデオの使用についての詳細は、A.2 節「ビデオ入出力方式の変更方法」を参照してください。

- 4) 赤と白のオーディオ・ケーブルを DVEVM オーディオ出力と LCD ディスプレイの R/L オーディオ入力ジャックに接続します（下記参照）。

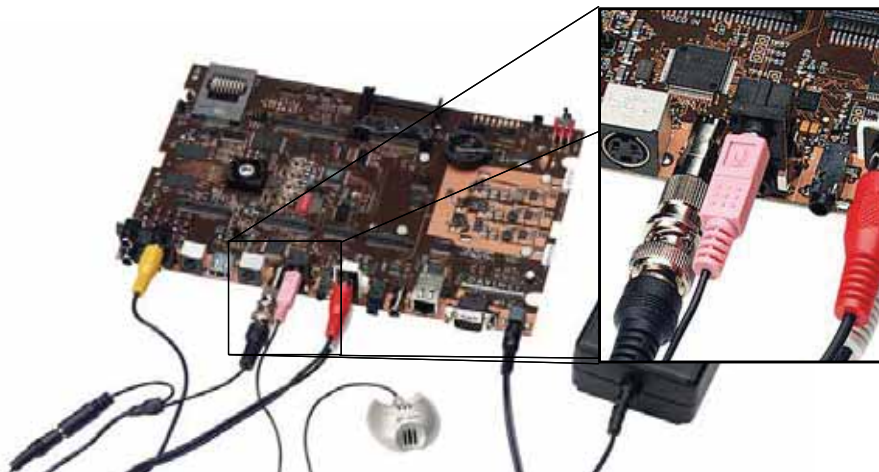


- 5) BNC-to-RCA コネクタを同軸ケーブルに接続します。次に、同軸ケーブルをビデオカメラと DVEVM ビデオ入力に接続します。
- 6) ビデオカメラの電源ジャックを接続します。静電放電（ESD）の影響を防止するために、この後の手順で差し込むように指示があるまでは、カメラの電源コードの反対側を差し込んではいけません。



S ビデオまたはコンポーネント・ビデオの使用についての詳細は、A.2 節「ビデオ入出力方式の変更方法」を参照してください。

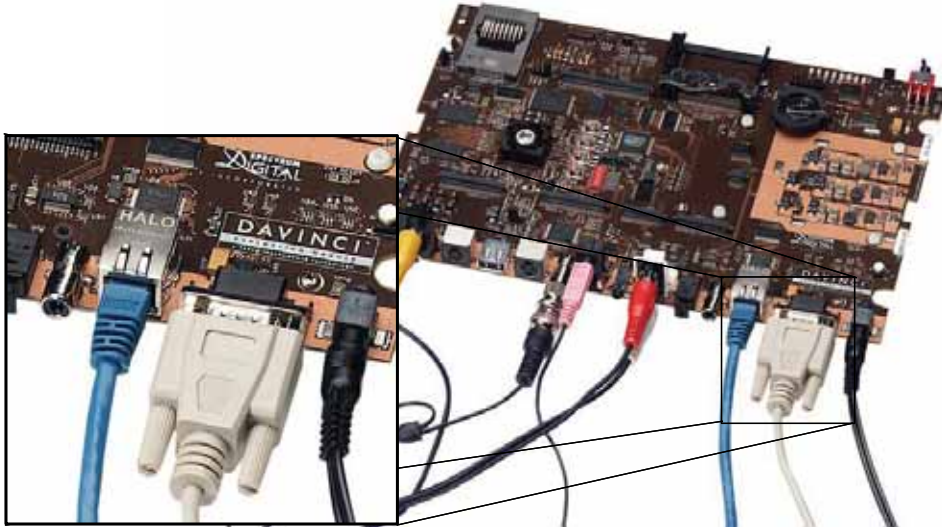
- 7) マイクフォンを DVEVM に接続します。
- 8) 電源ケーブルをボード上の DVEVM 電源ジャックに接続します。静電放電 (ESD) の影響を防止するために、まだカメラの電源コードの反対側を差し込んではいけません。



- 9) イーサネット接続を使用する場合には、イーサネット・ケーブルを DVEVM 上のイーサネット・ポートおよびイーサネット・ネットワーク・ポートに接続します。

ボードのイーサネット・コントローラを DHCP サーバが稼働しているコンピュータに接続しない場合には、ボードの起動に多少時間がかかることがあります。

- 10) コンソール・ウィンドウ用に UART ポートを使う予定がある場合には、RS232 スerial モデム・ケーブルを DVEVM UART ポートおよびホストの Linux ワークステーションの COM ポートに接続します。コンソール・ウィンドウの使用についての詳細は、2.2 節「コンソール・ウィンドウに接続する方法」を参照してください。



- 11) LCD ディスプレイの電源ケーブルを電源に差し込みます。
- 12) NTSC/PAL ビデオカメラの電源ケーブルを電源に差し込みます。
- 13) DVEVM ボードの電源ケーブルを電源に差し込みます。
- 14) LCD ディスプレイの電源を入れます。
- 15) DVEVM ボードの電源を入れます。
- 16) LCD ディスプレイには、デモ・ソフトウェアの初期画面が表示されます。赤外線リモート・コントロールを使って、ソフトウェアを実行します（第3章を参照）。

ネットワークに接続していない場合には、DHCP 関連の遅延があることに注意してください。

2.2 コンソール・ウィンドウに接続する方法

DVEVM のブート・メッセージの表示と割り込みを可能にするコンソール・ウィンドウをオープンすることができます。その手順は次のとおりです。

- 1) DVEVM のシリアルポートと PC のシリアル (COM) ポートをシリアルケーブルで接続します。
- 2) PC 上で HyperTerminal セッションを起動し、シリアルポートに接続するように構成します。
- 3) DVEVM の電源を入れると、ブートシーケンス・メッセージが表示されます。ブートシーケンスを中断するには、何かキーを押して、U-Boot コマンドシェルでコマンドを入力します。本書では、U-Boot コマンドシェルで入力するコマンドは、EVM # プロンプトで示します。

デモンストレーション・ソフトウェアの実行方法

本章では、DVEVM に付属しているデモンストレーション・ソフトウェアの実行方法について説明します。

項目	ページ
3.1 デフォルトのブート構成	3-2
3.2 スタンドアロン・デモの開始方法	3-2
3.3 スタンドアロン・デモの実行方法	3-3
3.4 ネットワーク・デモの実行方法	3-10

3.1 デフォルトのブート構成

DVEVM を箱から取り出してセットアップし、ボードの電源を入れると、しばらくしてデモが自動的に開始します。スタンドアロンでデモを実行するために、NFS マウントや TFTP サーバは必要ありません。デフォルトでは、DHCP サーバが必要ですが、ほとんどのルータが提供する DHCP サービスがあれば、デモを実行するには十分です。デフォルトでは、DHCP サーバから動的な IP アドレスを取得します。

標準ブートを中止するには、コンソール・ウィンドウで何かキーを押します (2.2 節を参照)。またブート構成を変更する場合には、A.5 節「その他のブート方法」も参照してください。

3.2 スタンドアロン・デモの開始方法

DVEVM ハードウェアを接続すると、事前にインストールされているデモ・プログラムが自動的に実行されて LCD ディスプレイに表示されます。デモではオーディオ、ビデオ、音声のエンコードとデコードが行われます。デモを行う方法は、2 種類あります。

- **スタンドアロン。**これは電源投入時のデフォルトのモードです。デフォルトのブート構成では、ワークステーションに接続しなくても、デモは自動的に実行します。
- **コマンドライン。**DVEVM をワークステーションに接続し、必要なソフトウェアをインストールする (4.3.1 項「ターゲット Linux ソフトウェアのインストール方法」を参照) と、ボードの Linux コマンドラインからデモを実行することができます (A.1 節「コマンドラインからデモを実行する方法」を参照)。

DVEVM ボードがブートすると、LCD ディスプレイにはリモート・コントロールの画像が表示されます。赤外線リモート・コントロールを使うと、デモを制御できます。

スタンドアロン・モードでは以下の手順でデモを実行します。

- 1) 電池が赤外線リモート・コントロールに確実に装着されていることを確認します。
- 2) 初期画面に赤外線リモート・コントロールの図が表示されます。これはスタンドアロン・デモを実行するために使います。少し時間をかけて、さまざまなボタンの機能を確認してください。
- 3) これは汎用のリモート・コントロールなので、DVEVM デモを実行するのに必要なコードが使えるように設定する必要があるかもしれません。設定を行うには、リモート・コントロールの赤い光が点灯するまで、「Code Search」ボタンを押し続けてください。次に「DVD」ボタンを押して、コードとして「020」と入力します。
- 4) TV または他のモードで、リモート・コントロールを誤って押してしまった場合には、「DVD」ボタンを押してリモート・コントロールを正しいモードに戻します。
- 5) リモート・コントロールが DVD+020 コードを受け付けない場合には、完全にリセットするために電池をはずし、Power キーを最低でも 1 分間押してから電池を再び装着してください。その後、手順 3 にあるようにリモート・コントロールをプログラムします。



3.3 スタンドアロン・デモの実行方法

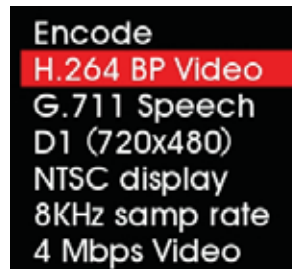
- 1) リモート・コントロールの図からメイン・メニュー画面に移動するために、リモート・コントロールの「Play」または「OK」を押します。次のような画面が表示されます。



Encode + Decode デモでは、ビデオの録画と再生を行うことができます。Encode デモでは、選択したフォーマットでオーディオ / 音声の録音およびビデオの録画を行います。Decode デモでは、選択したオーディオ / 音声ファイルおよびビデオファイルを再生します。Third Party Menu を使うと、さらにデモを追加することができます (A.3 節「デモ・アプリケーションを Third Party Menu に追加する方法」を参照)。

- 2) 上下矢印キーを使って、選択するデモを変更します。その後、「OK」または「Play」を押すと、選択したデモに切り替わります (この時点で「Power」を押すと、デモを完全に終了できます)。
- 3) デモの中では、設定画面から作業を開始します。デモを実行するために使用できる操作ボタンは画面の下部で確認し、現在の設定は画面の右上で確認します。

たとえば、Encode デモを使うと、ビデオをエンコードするときのビデオ・フォーマットおよびビットレートをセットすることができます。また組み込まれている設定をここに示します。

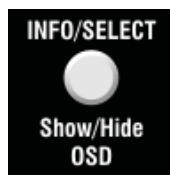


- 4) 上下矢印キーを使って、設定を変更したいメニューに移動します。



- 5) 左右の矢印キーを使って、変更したいメニュー設定が表示されるまでオプションを繰り返し移動します。
- 6) 「Play」を押して、Encode デモまたは Decode デモを開始します。「Rec」（録画）を 2 回押すと、Encode デモを開始します。「Stop」を押すと、メイン・メニューに戻ります。
- 7) デモの実行中には、設定、プロセッサの負荷、レートに関するデータが表示されます。静的な設定は右側に表示されます。レポートされる動的なデータは左側に表示されます。たとえば次のとおりです。

ARM CPU load:	7%	Encode
DSP CPU load:	89%	H.264 BP Video
Video frame rate:	30 fps	G.711 Speech
Video bit rate:	4050 kbps	D1 (720x480)
Audio bit rate:	61 kbps	NTSC display
Time elapsed:	00:00:24	8KHz samp rate



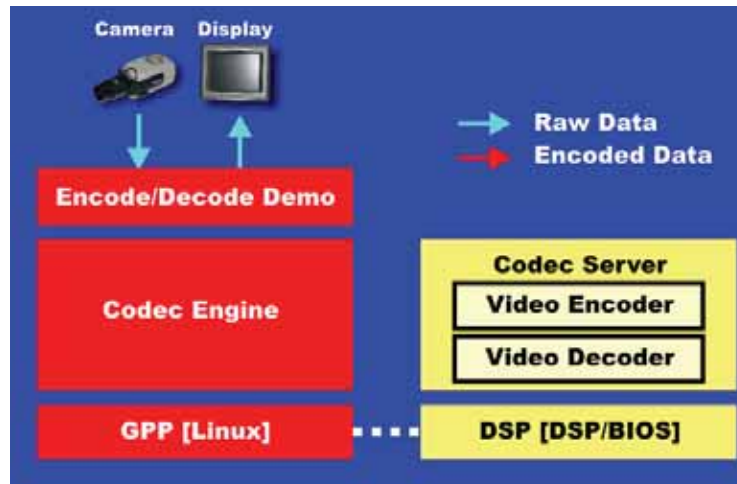
- 8) この情報はビデオ画面に重なって表示されます。そのため、表示されるビデオは実際のビデオより暗くなります。ビデオがもっとはっきりと見えるようにするために情報表示を非表示にするには、赤外線リモート・コントロールの「Info/Select」ボタンを押します。デモの実行中に、リモート・コントロールの左右の矢印キーを使うと、OSD（オーバーレイ）の透明性を変更できます。
- 9) デモを終了するか、または一時停止する場合は、「Stop」または「Pause」を押します。設定画面から「Stop」を押すと、メイン・メニューに戻ります。

デモはコーデック・エンジンを使って、GPP 側のアプリケーションが DSP 上で透過的に（意識することなく）アルゴリズムを実行できるようにしています。

DSP がアルゴリズムを実行中でないのに、最初から DSP CPU の負荷が高いことに気付くかもしれません。DSP がブートしている間は、DSP CPU の負荷は 100% から始まり、その後 CPP からの作業要求を待機している間に負荷は減少していきます。たとえ DSP がアイドル状態でも、CPU の負荷がゼロになるまでには多少時間がかかります（数秒）。というのも、コーデック・エンジンの CPU の負荷計算には、いろいろな処理に要した時間が含まれているからです。

3.3.1 Encode + Decode デモについて

Encode + Decode デモを実行すると、ビデオの録画と再生を行うことができます。ビデオカメラから入力されたデータは、エンコードされ、次にデコードされて、LCD ディスプレイに送られます。



Encode + Decode はビデオ処理だけを行っています。オーディオや音声のエンコードやデコードは行っていない。サポートされているビデオのアルゴリズムは、H.264 Baseline Profile（ファイル拡張子 .264）です。

表 3-1. Encode + Decode デモを行うための赤外線リモート・コントロールのボタン

赤外線リモート・コントロールのボタン	モード	動作
上下	--	-- 何もしない --
Play または OK	セットアップ	デモを開始します。
Record	--	-- 何もしない --
Info/Select	セットアップ	デモ用のブロック図の表示 / 非表示を切り換えます。
Info/Select	実行	情報表示の表示 / 非表示を切り換えます。
左右	実行	情報の透過レベルを変更します。
Pause	実行	デモを一時停止します（Play を押すと再開します）。
Stop	セットアップ / 実行	前の画面に戻ります。

アプリケーションは Linux を使用した ARM 上で動作しています。ビデオ信号は、コーデック・エンジンによって DSP 上のビデオ・エンコーダおよびデコーダに渡されます。データを渡す際に、共有メモリが使われます。

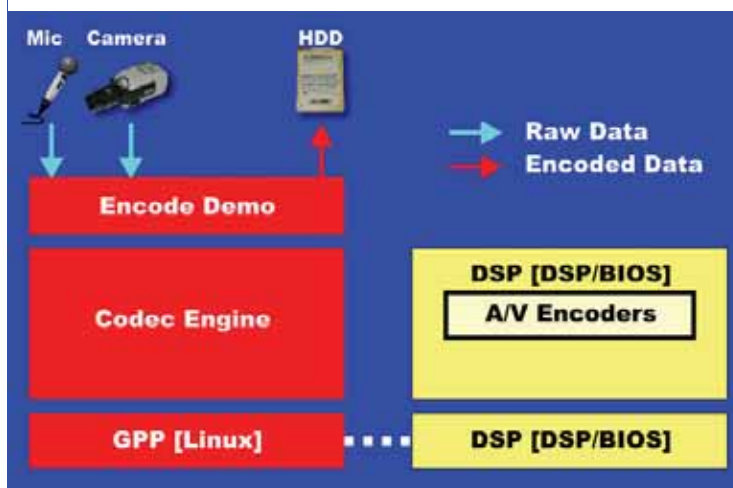
コマンドラインからこのデモを実行させるには、A.1.1 項「コマンドラインから実行する Encode/Decode デモ」を参照してください。

3.3.2 Encode デモについて

Encode + Decode デモと同様に、Encode デモもビデオをエンコードします。さらに、オーディオまたは音声もエンコードします。オーディオまたは音声のソースはマイクロフォンです。

エンコードされたデータは、DVEVM のハードディスク・ドライブの複数のファイルに書き込まれます。書き込まれるファイル名は demo.264、demo.mpeg4、demompeg4.g711、および demo264.g711 です。これらのファイルが存在する場合、必要に応じて上書きされます。

設定情報および負荷や各種レートに関して収集された動的なデータを表示する目的以外には出力はデコードされずに、LCD ディスプレイまたはスピーカーに送られません。



オーディオ・エンコーダではなく、音声エンコーダだけを使うことができます。ことに注目してください。サポートされているビデオ・アルゴリズムは、MPEG4（ファイル拡張子 .mpeg4）および H.264（ファイル拡張子 .264）です。サポートされている音声アルゴリズムは、G.711（ファイル拡張子 .g711）です。

表 3-2. Encode デモを行うための赤外線リモート・コントロールのボタン

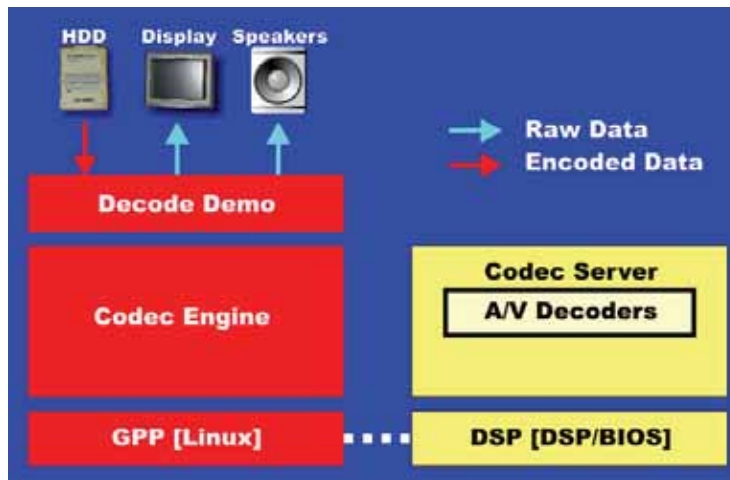
赤外線リモート・ コントロールのボタン	モード	動作
上下	セットアップ	オプション選択を変更します。
左右	セットアップ	選択したオプションの設定を変更します。
Play	セットアップ	Decode デモのセットアップに切り換えます。
Record (2 回) または OK	セットアップ / 実行	Encode デモを開始します。
Info/Select	セットアップ	デモ用のブロック図の表示 / 非表示を切り換えます。
Info/Select	実行	情報表示の表示 / 非表示を切り換えます。
左右	実行	情報の透過レベルを変更します (情報が前面に表示されているため、 Encode デモが表示されません)。
Pause	実行	デモを一時停止します (Play を押すと再開します)。
Stop	セットアップ / 実行	前の画面に戻ります。

アプリケーションは **Linux** を使用した **ARM** 上で動作しています。ビデオおよびオーディオ信号は、コーデック・エンジンによって **DSP** 上のエンコーダに渡されます。データを渡す際に、共有メモリが使われます。

コマンドラインからこのデモを実行させるには、**A.1.2** 項「コマンドラインから実行する **Encode** デモ」を参照してください。

3.3.3 Decode デモについて

Decode デモでは、選択したオーディオ / 音声ファイルおよびビデオファイルを再生します。デコードするビデオ・ファイルとオーディオ / 音声ファイルを選択します。左右の矢印ボタンを使って、DVEVM ハードディスク・ドライブに格納されている、デモに使うファイルと Encode デモで作成したファイルを選択します。デコードされた信号は、LCD ディスプレイおよびスピーカーに送られます。



サポートされているビデオ・アルゴリズムは、MPEG4（ファイル拡張子 .mpeg4）、H.264（ファイル拡張子 .264）、および MPEG4（ファイル拡張子 .mpeg4）です。

サポートされているオーディオ・アルゴリズムは、AAC（ファイル拡張子 .aac）および MPEG1 レイヤ 2（ファイル拡張子 .mp2）です。サポートされている音声アルゴリズムは、G.711（ファイル拡張子 .g711）です。

表 3-3. Decode デモを行うための赤外線リモート・コントロールのボタン

赤外線リモート・コントロールのボタン	モード	動作
上下	--	-- 何もしない --
左右	セットアップ	別のファイルの組み合わせを選択します。
Play または OK	セットアップ	Decode デモを開始します。
Record	--	-- 何もしない --
Info/Select	セットアップ	デモ用のブロック図の表示 / 非表示を切り換えます。

表 3-3. Decode デモを行うための赤外線リモート・コントロールのボタン（続き）

赤外線リモート・ コントロールのボタン	モード	動作
Info/Select	実行	情報表示を切り換えます。
左右	実行	情報の透過レベルを変更します。
Pause	実行	デモを一時停止します（Play を押すと再開します）。
Stop	セットアップ / 実行	前の画面に戻ります。

アプリケーションは Linux を使用した ARM 上で動作しています。ビデオおよびオーディオ信号は、コーデック・エンジンによって DSP 上のデコーダに渡されます。データを渡す際に、共有メモリが使われます。

コマンドラインからこのデモを実行させるには、A.1.3 項「コマンドラインから実行する Decode デモ」を参照してください。

3.3.4 Third Party Menu について

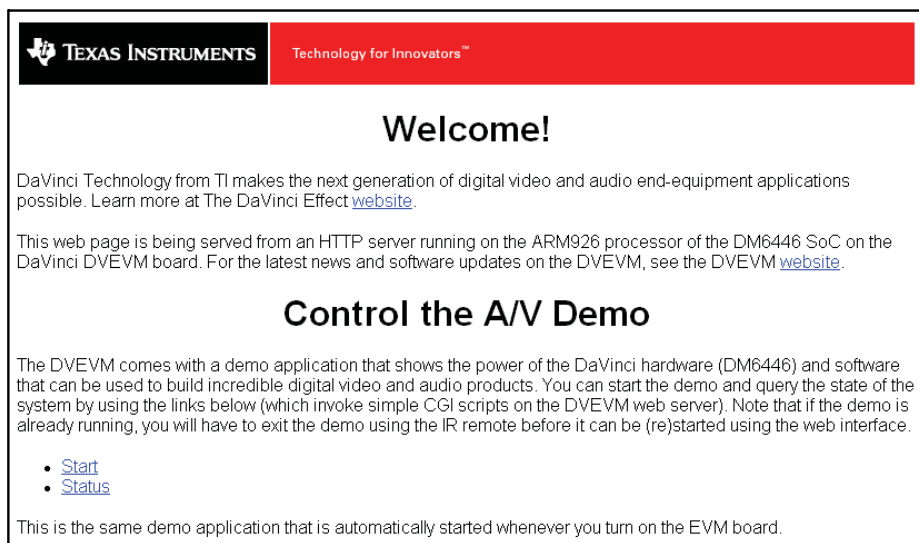
Third Party Menu を使うと、さらにデモを追加することができます。A.3 節「デモ・アプリケーションを Third Party Menu に追加する方法」を参照してください。

3.4 ネットワーク・デモの実行方法

標準 TCP/IP ネットワーク・サポートの例として、DVEVM には、小規模の HTTP Web サーバの例が含まれています。この Web サーバは、GPP 側で Linux のスタートアップ・シーケンスの一部として起動されます。標準 TCP/IP ポート 80 で Web ブラウザからのリクエストに応えるように設定されています。

DVEVM ボードのブート後に、PC を DVEVM ボードの接続されている同一ネットワークに接続します。「<http://ip-address-of-dvevm>」という形式の URL を Web ブラウザ (Internet Explorer、Firefox、Opera) に入力します。A/V デモのメイン・メニューの右下隅に、ボードの IP アドレスが表示されます。

DaVinci テクノロジーおよび DVEVM ソフトウェアに関する情報を含む Web ページが表示されます。



この Web ページを使用して、ボードと対話形式で A/V デモを実行します (3.3 節「スタンドアロン・デモの実行方法」を参照)。DVEVM の 2 つのわかりやすい CGI スクリプトを使うと、(デモがまだ実行されていないという前提で) デモを開始して、ボード上でどのようなプロセスが実行されているか確認することができます。デモが Web ページから起動されることを確認する場合は、最初にデモを終了させてください (メイン・メニューから Power ボタンを使用します)。

Web サーバ・ソフトウェアは、THHTTPD (<http://www.acme.com/software/thhttpd/>) というオープンソース・パッケージです。THHTTPD は、サイズが小さく、高速で、移植性が高くなるように設計されています。このソース・コードは、DVEVM ソフトウェアに付属しています。Web サイトから最新版を直接ダウンロードすることができます。Web サーバと CGI スクリプトは、ターゲットの `/opt/dvevm/web` ディレクトリにインストールされています。

DVEVM ソフトウェアのセットアップ

本章では、DVEVM に付属しているソフトウェアの使用方法について説明します。

項目	ページ
4.1 ソフトウェアの概要	4-2
4.2 インストールの準備	4-5
4.3 ソフトウェアのインストール方法	4-6
4.4 ビルド / 開発環境のセットアップ方法	4-11
4.5 ターゲットに合わせて DVEVM ソフトウェアを再ビルドする方法	4-12
4.6 新しい Linux カーネルのビルド方法	4-12
4.7 新しい Linux カーネルのブート方法	4-14
4.8 アップグレードのインストール	4-14

4.1 ソフトウェアの概要

アプリケーションの開発を始めるには、DVEVM 開発環境をインストールする必要があります。ここでは、DVEVM ソフトウェアを開発用のホストにロードする際に必要な手順の概要を説明します。まず始めに、3 枚の DVEVM 配布 CD またはその中のファイルが必要になります。

DaVinci ソフトウェア・アプローチは、相互運用性があり、最適化されたすぐに使えるビデオ・コーデックとオーディオ・コーデックを提供します。これは DSP と内蔵アクセラレータを利用します。これらのコーデックは、構成可能なフレームワークに組み込まれていて、ソフトウェアが迅速に開発できるように一般的なオペレーティング・システム（Linux など）用の公開 API として提供されます。

DVEVM は ARM 側の開発を行います。開発者は DSP 側を API セットを介してアクセス可能なブラック・ボックスとして扱います。DSP 側を直接プログラムする場合は、DVSDK（Digital Video Software Development Kit）を利用してください。

DVEVM には以下のソフトウェアが付属しています。

□ スタンドアロン・デモンストレーション・ソフトウェア

このソフトウェアは、DVEVM のハードディスク・ドライブ上にあります。このデモではオーディオ、ビデオ、音声のエンコードとデコードを行います。別のデモでは、ボードのネットワーク機能を示します。3.2 節「スタンドアロン・デモの開始方法」を参照してください。

□ CD 1: Getting Started Guide

この CD には、デモ・アプリケーション、コーデック・エンジン・ソフトウェア、コーデック・サーバー例、および DVEVM の資料が収められています。次のファイルが含まれます。

- sprue66b.pdf（本書）
- dvevm_setuplinux_#_##_##_#.bin
- mvl_lsp_setuplinux_#_##_##_#.bin
- restore ディレクトリ（ハードディスク・ドライブのリカバリ用のファイルが含まれます。詳細は、TI Technical Worldwide Support にお問い合わせください。）

□ CD 2: MontaVista Linux Pro v4.0 ターゲット・ファイル・システム

DVEVM に付属のバージョンは、正式リリース前のデモ版です。次のファイルが含まれます。

- mvl_target_setuplinux_#_##_##_#.bin
このインストール・ファイルには、MontaVista のターゲット・ファイル・システムが含まれます。

❑ **CD 3: MontaVista Linux Pro v4.0 システム・ツール**

DVEVM に付属のバージョンは、正式リリース前のデモ版です。次のファイルが含まれます。

■ **mv_l_setuplinux_#_##_###_###.bin**

このインストール・ファイルには、MontaVista ツールの開発ツール・チェーンが含まれます。

❑ **CD 4: A/V データ**

A/V サンプル・データが data.tar.gz 内に含まれます。

Texas Instruments は、MontaVista Software Inc. の同意のもと、Linux Professional Edition v4.0 組み込みオペレーティング・システムと開発ツールのデモ版を提供しています。DVEVM 基本キットには、このデモ版の正式リリース前のリリースが組み込まれています。このデモ版は、MontaVista が提供する完全版の Professional Edition のサブセットです。DevRocket™ などのツールや Professional Edition の資料は含まれていませんが、その他の部分ではお客様が DaVinci プラットフォームを評価するには十分な機能があり、利用価値があります。また、このリリースには MontaVista のユーザ・ライセンスは含まれていないため、MontaVista Software Inc. からの直接的なお客様サポート、保証、または損害賠償は提供されません。

このデモ版 MontaVista Linux の製品リリースが含まれている DVSDK を注文することができます。DVSDK には、完全な MontaVista ライセンスと DevRocket IDE が含まれます。

4.1.1 本書のコマンド・プロンプト

本書では、コマンドは入力される環境を示すプロンプトの後に示されます。

❑ **host \$**

ホスト Linux ワークステーションのシェル・ウィンドウにコマンドを入力することを表します。

❑ **EVM #**

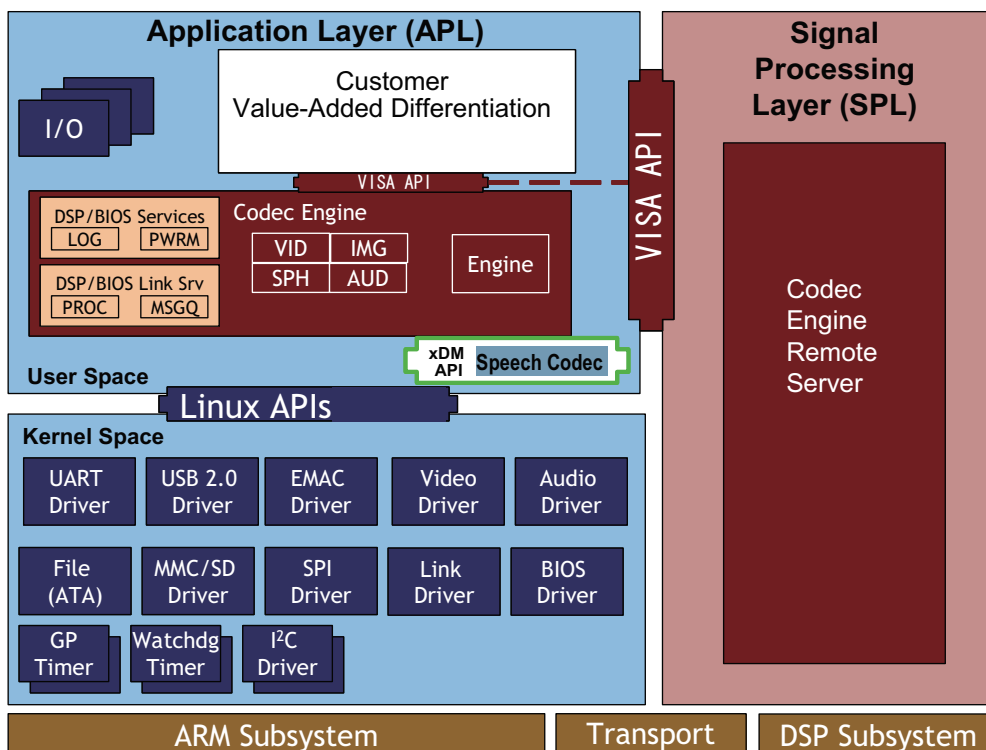
EVM ボードのシリアル・ポートに接続されたコンソール・ウィンドウ内の U-Boot シェルにコマンドを入力することを表します。(2.2 節)

❑ **target \$**

EVM ボードのシリアル・ポートに接続された端末ウィンドウ内の Linux シェルにコマンドを入力することを表します。

4.1.2 ソフトウェア・コンポーネント

次の図は、DVEVM でのアプリケーション開発に使用するソフトウェア・コンポーネントを示します。



上記の図では、ユーザのアプリケーションは ARM 側で実行されます。入出力 (I/O) とアプリケーションの処理を扱います。ビデオ、画像、音声、オーディオ信号を処理するために、コーデック・エンジンに付属の VISA API を使用します。次にコーデック・エンジンは、DSP/BIOS Link などのサービスと xDAIS や xDM などのプロトコルを使用して、DSP 側で事前構成済みのコーデック・エンジン・リモート・サーバと通信します。DSP は信号処理を扱い、その結果は共有メモリに置かれ、ARM 側で利用されます。詳細は、『Codec Engine Application Developer's Guide』(資料番号 SPRUE67) を参照してください。

また、ARM 側の Linux では、ユーザ・アプリケーションは多くの API を使用可能です。これにはドライバとタイマーが含まれます。

4.2 インストールの準備

ホスト・システム上で、3 枚の DVEVM デモンストレーション CD をマウントし、次の `.bin` ファイルを 1.2 GB 以上の空き容量がある一時的な場所にコピーします。インストール・ファイルはソフトウェアをインストールした後に削除できるため、`/tmp` などのディレクトリを推奨します。

- ☐ `mv1_setuplinux_#_##_##_#.bin`
- ☐ `mv1_target_setuplinux_#_##_##_#.bin`
- ☐ `mv1_lsp_setuplinux_#_##_##_#.bin`
- ☐ `dvevm_setuplinux_#_##_##_#.bin` または

X グラフィカル画面が使用可能であることを確認し、`DISPLAY` 環境変数を次の値に指定します。たとえば次のとおりです。

`csh`:

```
host $ setenv DISPLAY cnabc0314159d1:0
```

`ksh`:

```
host $ export DISPLAY=cnabc0314159d1:0
```

4.3 ソフトウェアのインストール方法

DVEVM で使用されるソフトウェアを以下の手順でインストールします。

- ❑ 4.3.1 項「ターゲット Linux ソフトウェアのインストール方法」
- ❑ 4.3.2 項「DVEVM ソフトウェアのインストール方法」
- ❑ 4.3.3 項「A/V デモ・ファイルのインストール方法」
- ❑ 4.3.4 項「ターゲットからのアクセス用に共有ファイル・システムをエクスポートする方法」
- ❑ 4.3.5 項「共有ファイル・システムのテスト」
- ❑ 4.3.6 項「PAL ビデオ・ユーザ用のブート・セットアップの構成」

4.3.1 ターゲット Linux ソフトウェアのインストール方法

ここでは、ターゲット・ボードで使用する Linux をインストールする方法について説明します。これは MontaVista Linux Pro v4.0 のデモ版です。

ターゲットとユーザのホストの Linux ワークステーションでは、別々のバージョンの Linux が使用されることに注意してください。次の Linux ホスト・オペレーティング・システムを DVEVM と一緒に使用することができます。

- ❑ Red Hat Enterprise Linux v3
- ❑ Red Hat Enterprise Linux v4
- ❑ SuSe v10.0 Workstation

Linux ソフトウェアをインストールする手順は、次のとおりです。

- 1) **root** でホスト Linux ワークステーションにログインします。これにより、ユーザは MontaVista Linux をインストールするためのグラフィカル・インストーラを正しく実行できます。
- 2) Linux ツール、Linux カーネル、およびファイル・システムをインストールするためにコピーされた一時的な場所から次の bin ファイルを 1 つずつ実行します（ここで **###_###_###** は現行のバージョン番号です）。このインストール手順では、デフォルトのインストール・ディレクトリである /opt を使用することを想定しています。

```
host $ ./mvl_setuplinux_###_###_###.bin
host $ ./mvl_target_setuplinux_###_###_###.bin
host $ ./mvl_lsp_setuplinux_###_###_###.bin
```

これらのファイルから InstallShield を起動するには、1 ファイルにつき数分かかる場合があります。

- 3) これらのファイルを実行したら、次のファイルが `/opt/mv_pro_4.0`（またはデフォルトの代わりに選択したディレクトリの `/mv_pro_4.0` サブディレクトリ）にあることを確認してください。

- `mvlttools4.0-no-target.tar.gz`
- `mv14.0-target_path.tar.gz`
- `DaVinciLSP###.tar.gz`

- 4) `tar` ファイルを解凍する場所に移動します。たとえば次のとおりです。

```
host $ cd /opt/mv_pro_4.0
```

- 5) `tar` ファイルを（root で）次のコマンドを使用して解凍します。

```
host $ tar xzf mvlttools4.0-no-target.tar.gz
host $ tar xzf mv14.0-target_path.tar.gz
host $ tar xzf DaVinciLSP###.tar.gz
```

これで MontaVista ディレクトリ構造が `/opt/mv_pro_4.0/montavista/` ディレクトリ下に作成されます。

4.3.2 DVEVM ソフトウェアのインストール方法

DVEVM ソフトウェアには、コーデック・エンジン・コンポーネント、DSP/BIOS Link、サンプル・データ・ファイル、xDAIS と xDM のヘッダ・ファイル、および Linux 用連続メモリ・アロケータ（CMEM）が含まれています。

- ❑ コーデック・エンジンは、A/V コーデックの作成と相互作用のためのフレームワークを提供します。これらの A/V コーデックは、ARM 上で C 呼び出し可能な Linux API を用いての xDM インターフェイスの反映として DSP で実行されるものです。
- ❑ DSP/BIOS Link は、GPP/DSP インターフェイスを提供します。このインターフェイスは、コーデック・エンジンが Linux から DSP を制御して DSP と通信するために使用するものです。

Linux インストーラを使用して DVEVM ソフトウェアを次の手順でインストールします。

- 1) ユーザ・アカウントを使用してログインします。以降の手順では、ユーザのホーム・ディレクトリを「`~`」と示します。
- 2) DVEVM CD からソフトウェアをインストールします。たとえば次のとおりです。

```
host $ cd /tmp
host $ ./dvevm_setuptoolslinux_1_00_00_bb.bin
```

- 3) プロンプトが表示されたとき、デフォルトのインストール場所を使用しないでください。その代わりに、使用しているアカウントのホーム・ディレクトリにソフトウェアをインストールしてください。

たとえば、ホーム・ディレクトリが `/home/useracct` の場合、そのディレクトリをインストール場所のダイアログに入力します。DVEVM ソフ

トウェアは `/home/useracct/dvevm_#_##` 下にインストールされます。この `_##` はバージョン番号です。

- 4) これで一時的な場所に置いた `.bin` ファイルを削除することができます。

注: ディレクトリ上で `rm -rf` コマンドを使用すると、これらのコンポーネントのいずれかをアンインストールすることができます。
`InstallShield` で作成された `_uninstall` ディレクトリは無視してください。

DVSDK にバンドルされる **Monta Vista Linux** のライセンス版をすでにお持ちのお客様は、**Microsoft Windows** インストーラが提供されています。ご使用のバージョンの **Monta Vista Linux** に対応したインストーラを使用してください。詳細は、A.6 節「**Windows** でのコンポーネントのインストール方法」を参照してください。

4.3.3 A/V デモ・ファイルのインストール方法

4 枚目の CD には、デモで使用する A/V ファイルが収められています。前の節の手順を実行したら、以下の手順を実行して A/V ファイルをインストールします。

- 1) 上記でセットアップした DVEVM ディレクトリに移動します。たとえば次のとおりです。

```
host $ cd ~/dvevm_1_xx
```

- 2) A/V データ CD をマウントし、ファイルをユーザの DVEVM ディレクトリにコピーします。たとえば次のとおりです。

```
host $ cp /mnt/cdrom/data.tar.gz .
```

- 3) A/V データ・ファイルを解凍します。たとえば次のとおりです。

```
host $ tar xzf data.tar.gz
```

4.3.4 ターゲットからのアクセス用に共有ファイル・システムをエクスポートする方法

ボードのハードディスク・ドライブにはファイル・システムが組み込まれていますが、開発ではターゲット・ボードの NFS をホスト **Linux** ワークステーションのファイル・システムにマウントする方が便利です。アプリケーションをテストしたら、それをスタンドアロン・デモンストレーション用にボードのハードディスク・ドライブに保管できます。

ボードでターゲット・ファイル・システムをマウントするには、事前にホスト **Linux** ワークステーションでそのターゲット・ファイル・システムをエクスポートしておく必要があります。このファイル・システムは、**NFS** (**Network File System**) サーバを使用します。エクスポートされたファイル・

システムには、ターゲット・ファイル・システムとユーザの実行可能ファイルが含まれます。

次の手順で NFS サーバからファイル・システムをエクスポートしてください。この手順は一度だけ実行する必要があります。

- 1) ユーザ・アカウントでホスト Linux ワークステーションにログインします。以降の手順では、ユーザのホーム・ディレクトリを「~」と示します。

- 2) 次のコマンドを実行して、MontaVista ファイル・システムの場所を用意します。

```
host $ cd ~
host $ mkdir -p workdir/filesys
host $ cd workdir/filesys
```

- 3) ホスト Linux ワークステーションでユーザから「root」に切り替えます。

```
host $ su root
```

- 4) 次のコマンドを実行して、<useracct> で共有領域に書き込むための権限をもつターゲット・ファイル・システムのコピーを作成します。<useracct> をユーザ名に置き換えてください。/opt/mv_pro_4.0 以外の場所にインストールした場合は、cp コマンドでユーザの場所を使用します。

```
host $ cp -a /opt/mv_pro_4.0/montavista/pro/devkit/arm/v5t_le/target/* .
host $ chown -R <useracct> opt
```

- 5) ホスト Linux ワークステーションで /etc/exports ファイルを編集します。filesys 領域のエクスポート用に次の行を追加します。<useracct> をユーザ名に置き換えてください。ルート・ディレクトリからの絶対パスを使用してください。~ はファイル・システムによってはエクスポートに対して機能しない場合があります。

```
/home/<useracct>/workdir/filesys *(rw,no_root_squash,no_all_squash,sync)
```

- 6) root として次のコマンドを使用して、NFS サーバに構成の変更を認識させ、次に NFS を再起動させます。

```
host $ /usr/sbin/exportfs -a
host $ /sbin/service nfs restart
```

4.3.5 共有ファイル・システムのテスト

NFS セットアップをテストするには、次の手順を踏んでください。

- 1) ホスト Linux ワークステーションの IP アドレスを次のようにして取得します。eth0 イーサネット・ポートに関連付けられている IP アドレスを検索します。

```
host $ /sbin/ifconfig
```

- 2) 端末エミュレーション・ウィンドウを開き、RS-232 経由で DVEVM ボードに接続します。Windows ワークステーションの場合は、HyperTerminal を使用できます。Linux ワークステーションの場合は、Minicom を使用できます。

- 3) DVEVM ボードの電源を入れ、コンソール・ウィンドウでキーを押して自動ブートシーケンスを終了させます (2.2 節)。
- 4) コンソール・ウィンドウで、次の環境変数をセットします。

```
EVM # setenv nfshost <IP ADDRESS OF YOUR NFS HOST>
EVM # setenv rootpath <DIRECTORY TO MOUNT>
EVM # setenv bootargs console=ttyS0,115200n8 noinitrd rw
      ip=dhcp root=/dev/nfs
      nfsroot=$(nfshost):$(rootpath),nolock mem=120M
```

- 5) この環境を保存して、EVM ボードの電源を入れるたびにこれらのコマンドを再入力する必要がないようにします。

```
EVM # saveenv
```

- 6) NFS を使用してボードをブートします。

```
EVM # boot
```

TFTP または NFS によるブート方法、およびフラッシュまたはボードのハードディスク・ドライブの使用方法については、A.5 節「その他のブート方法」を参照してください。

4.3.6 PAL ビデオ・ユーザ用のブート・セットアップの構成

デフォルトのブートシーケンス中に NTSC または PAL のいずれかのビデオ規格を選択するように DVEVM を構成できます。PAL を選択するには、S3 (USER) スイッチ・バンクのスイッチ 10 をオンにセットします。NTSC の場合は、このスイッチをオフにセットします。このスイッチにより、U-Boot 環境変数「videostd」が「pal」または「ntsc」にセットされます。

Linux カーネルに渡される「bootargs」環境変数の「videostd」変数を使用すると、対応するビデオ標準がディスプレイ (VPBE) ドライバで使用されます。デフォルトの「bootcmd」環境変数は、このタスクを次のように実行します。

```
bootargs="mem=120M console=ttyS0,115200n8 root=/dev/hda1 rw
noinitrd ip=dhcp"

bootcmd="setenv setboot setenv bootargs \$(bootargs)
video=dm64xxfb:output=\$(videostd);run setboot;bootm
0x2050000"
```

「boot」コマンドが U-Boot で実行される場合、「videostd」の値はスイッチの設定に基づいて置き換えられます。

本書のほかの部分にある bootargs の例は、デフォルトの NTSC ビデオ出力が使用されることを想定しています。PAL ビデオを使用している場合は、それに応じてこれらの例を変更する必要があります。

S-Video とコンポーネント・ビデオへの切り替えについては、A.2 節「ビデオ入出力方式の変更方法」を参照してください。

4.4 ビルド / 開発環境のセットアップ方法

次の手順で GPP 側の開発とビルドの環境をセットアップしてください。

- 1) NFS ホスト・システムの**ユーザ**・アカウントにログインします (root ではありません)。
- 2) MontaVista ツール・チェーンのホスト・ツールとクロス・コンパイラ (arm_v5t_le-gcc) を検出できるように PATH をセットします。たとえば、MontaVista LSP のデフォルトのインストールでは、次のような定義をユーザのシェル・リソース・ファイル (~/bashrc など) に追加します。/opt/mv_pro_4.0 以外の場所にインストールした場合は、その位置を PATH で指定します。

```
PATH="/opt/mv_pro_4.0/montavista/pro/devkit/arm/v5t_le/bin:
/opt/mv_pro_4.0/montavista/pro/bin:
/opt/mv_pro_4.0/montavista/common/bin:$PATH"
```

4.4.1 簡単なプログラムを作成して DVEVM 上で実行する方法

4.3.4 項「ターゲットからのアクセス用に共有ファイル・システムをエクスポートする方法」および 4.4 節「ビルド / 開発環境のセットアップ方法」の手順を実行したことを確認してください。

NFS ホスト・システム上で、次の手順をユーザとして実行します (root ではありません)。

- 1) host \$ **mkdir ~/workdir/filesys/opt/hello**
- 2) host \$ **cd ~/workdir/filesys/opt/hello**
- 3) 次の内容の hello.c というファイルを作成します。

```
#include <stdio.h>

int main() {
    printf("Buongiorno DaVinci!\n");
    return 0;
}
```

- 4) host \$ **arm_v5t_le-gcc hello.c -o hello**

ターゲット・ボード上で、次の手順を実行します。ターゲットのコンソール・ウィンドウ (2.2 節) または telnet セッションのいずれかを使用できます。

- 1) target \$ **cd /opt/hello**
- 2) **./hello** を実行します。出力は次のようになります。

```
Buongiorno DaVinci!
```

4.5 新しい Linux カーネルのビルド方法

ターゲットの Linux カーネルのソースを変更する場合は、そのカーネルを再ビルドする必要があります。その後、DVEVM ボードのフラッシュにインストール済みのカーネルを置き換えるか、またはネットワーク接続上でカーネルをブートするように U-Boot ユーティリティで TFTP を使用するかしてこのカーネルをブートします。

新しいカーネルをビルドする場合は、事前に 4.4 節「ビルド / 開発環境のセットアップ方法」および 4.4.1 項「簡単なプログラムを作成して DVEVM 上で実行する方法」を完了していることを確認してください。ユーザの PATH に少なくとも MontaVista ツールが含まれている必要があります。

```
PATH="/opt/mv_pro_4.0/montavista/pro/devkit/arm/v5t_le/bin:
/opt/mv_pro_4.0/montavista/pro/bin:$PATH"
```

次の手順で Linux カーネルを再ビルドします。

- 1) ユーザ・アカウントにログインします (root ではありません)。
- 2) 次のようなコマンドを使用して、ホーム・ディレクトリに MontaVista Linux Support Package (LSP) のローカル作業コピーを作成します。このコピーには、組み込み Linux 2.6.10 カーネルと DaVinci ドライバが含まれます。/opt/mv_pro_4.0 以外の場所にインストールした場合は、cp コマンドでユーザの場所を使用します。

```
host $ cd ~
host $ mkdir -p workdir/lsp
host $ cd workdir/lsp
host $ cp -R /opt/mv_pro_4.0/montavista/pro/devkit/lsp/ti-davinci .
```

- 3) 次のコマンドを使用して、DaVinci のデフォルトを使用するカーネルを構成します。CROSS_COMPILE はコンパイル時に使用される実行ファイルに対してプレフィックスを指定することにご注意ください。

```
host $ cd ti-davinci
host $ make ARCH=arm CROSS_COMPILE=arm_v5t_le- davinci_dm644x_defconfig
```

- 4) カーネル・オプションを変更するには、「make menuconfig」コマンドを使用する必要があります。このコマンドの方法については、MontaVista の資料を参照してください。MontaVista のデフォルトのカーネル・オプションをイネーブルにするには、次のコマンドを使用します。

```
host $ make ARCH=arm CROSS_COMPILE=arm_v5t_le- checksetconfig
```

- 5) 次のコマンドを使用して、カーネルをコンパイルします。

```
host $ make ARCH=arm CROSS_COMPILE=arm_v5t_le- uImage
```

- 6) 次のコマンドを使用して、U-Boot が TFTP で EVM にダウンロードできる場所に uImage をコピーします。これらのコマンドは、ユーザがデフォルトの TFTP ブート領域 /tftpboot を使用することを想定しています。

す。他の TFTP ルートの場所を使用する場合は、/tftpboot をユーザ指定の TFTP ルートの場所に変更してください。

```
host $ cp ~/workdir/lsp/ti_davinci/arch/arm/boot/uImage /tftpboot
host $ chmod a+r /tftpboot/uImage
```

Linux のビルド構成オプションの詳細については、標準 Linux カーネルの参考資料またはオンライン・ソースを参照してください。

4.6 ターゲットに合わせて DVEVM ソフトウェアを再ビルドする方法

デモ・ファイルを /opt/dvevm ディレクトリに置くには、DVEVM ソフトウェアを再ビルドする必要があります。この場合は、次の手順を実行します。

- 1) ディレクトリを ~/dvevm_#_## に変更します。
- 2) ~/dvevm_#_###/Rules.make ファイルを編集します。EXEC_DIR が NFS エクスポート済みファイル・システムの opt ディレクトリを次のように指定していること、および LINUXKERNEL_INSTALL_DIR が次のように定義されていることを確認します。

```
EXEC_DIR=/home/<useracct>/workdir/filesys/opt/dvevm
LINUXKERNEL_INSTALL_DIR=/opt/mv_pro_4.0/montavista/pro/devkit/lsp/ti-davinci
```

- 3) Rules.make が入っている同じディレクトリ内で、次のコマンドを使用して DVEVM 製品をビルドし、生成されたバイナリをターゲット・ファイル・システム (/opt/dvevm など) に置きます。

```
host $ make
host $ make install
```

4.7 新しい Linux カーネルのブート方法

新しいカーネルをビルドしたら、それを使用して DaVinci ボードをブートするために TFTP 経由でボードに転送する必要があります。ここでは、4.5 節「新しい Linux カーネルのビルド方法」の手順を完了していること、およびブート・ファイル `uImage` が `/tftpboot`（または他のサイト固有の TFTP でアクセス可能な場所）にコピーされていることを想定しています。

- 1) DVEVM ボードの電源を入れ、コンソール・ウィンドウでキーを押して自動ブートシーケンスを終了します (2.2 節)。
- 2) 次の環境変数をセットします（この場合、ユーザがデフォルトの何も変更されていない U-Boot 環境から起動することを想定しています）。
U-Boot のデフォルト環境については、3.1 節「デフォルトのブート構成」を参照してください。

```
EVM # setenv bootcmd 'dhcp;bootm'
EVM # setenv serverip <YOUR TFTP SERVER IP ADDRESS>
EVM # setenv bootfile uImage
EVM # setenv bootargs mem=120M console=ttyS0,115200n8
      root=/dev/hda1 rw noinitrd ip=dhcp
```

この構成により、新しい Linux カーネルが TFTP を介してハードディスク・ドライブを使用したファイル・システムでブートします。NFS ファイル・システムと TFTP を組み合わせて使用する方法については、A.5.4 項「NFS ファイル・システムを使用して TFTP 経由でブートする方法」を参照してください。

4.8 アップグレードのインストール

アップグレード・パッケージからソフトウェアをインストールする場合は、そのパッケージに付属の手順に従ってください。

追加手順

この付録では、ユーザのセットアップ内容や特定の目的に応じて使用できるオプションの手順について説明します。

項目	ページ
A.1 コマンドラインからデモを実行する方法	A-2
A.2 ビデオ入出力方式の変更方法	A-6
A.3 デモ・アプリケーションを Third Party Menu に追加する方法	A-9
A.4 TFTP サーバのセットアップ方法.....	A-11
A.5 その他のブート方法	A-12
A.6 Windows でのコンポーネントのインストール方法	A-16
A.7 DSP/BIOS Link の再ビルド方法	A-17
A.8 DVEVM ハードディスク・ドライブの復元および更新方法	A-18

A.1 コマンドラインからデモを実行する方法

EVM ボードのシリアル・ポートに接続された端末ウィンドウ内の Linux シェルからデモ・アプリケーションを実行することができます。これらのデモは、3.2 節「スタンドアロン・デモの開始方法」で説明しているものと同一です。

デモ・アプリケーションのコマンドライン構文とオプションは、以降の項に記載されています。

コマンドラインからデモ・アプリケーションを実行するときは事前に、DSP/BIOS Link と CMEM カーネルのモジュールをロードしておく必要があります。これらのモジュールをロードするには、デモが入っているディレクトリから次のコマンドを使用します。

```
Target $ ./loadmodules.sh
```

サポートされている A/V ファイルのファイル拡張子は次のとおりです。

- ☐ ビデオ。H.264 Baseline Profile (*.264 ファイル)
- ☐ ビデオ。MPEG4 (*.mpeg4 ファイル)
- ☐ ビデオ。MPEG2 (*.m2v ファイル)
- ☐ 音声。G.711 (*.g711 ファイル)
- ☐ オーディオ。AAC (*.aac ファイル)
- ☐ オーディオ。MPEG1 レイヤ 2 (*.mp2 ファイル)

A.1.1 コマンドラインから実行する Encode/Decode デモ

このデモは、コーデック・エンジンを使用して、H.264 アルゴリズムを使用したキャプチャ・デバイス (V4L2) からデータを取得して中間バッファにエンコードします。その後、データは表示フレーム・バッファにデコードされます。

```
Target $ ./encodedecode [options]
```

オプション:

- ☐ **-r|--resolution**
デモの解像度。CIF、ZOOM、または D1 のいずれかを指定します。デフォルトは D1 です。
 - CIF。NTSC システムでは 352x240、PAL システムでは 352x288 でキャプチャして表示します。画像は画面の中央に配置されます。
 - ZOOM。NTSC システムでは 352x240、PAL システムでは 352x288 でビデオをキャプチャします。データのエンコードとデコードが完了すると、データはディスプレイにより D1 解像度 (NTSC では 720x480、PAL では 720x576) にズームされます。
 - D1。NTSC システムでは 720x480、PAL システムでは 720x576 でキャプチャして表示します。

- ❑ **-t|--time**
デモを実行する秒数。デフォルトでは、時間制限はありません。
- ❑ **-i|--interface**
これを使用すると、メイン・デモ・インターフェイスがこのデモの終了時に起動します。デフォルトでは、オフになっています。
- ❑ **-x**
S-Video 入力フォーマットを選択します。このフラグは、DVEVM 1.1 以降でのみ使用できます (DVEVM 1.0 はコンポジット・ビデオのみをサポートします)。
- ❑ **-h|--help**
このヘルプ・メッセージを出力します。

次の例は ZOOM 解像度を使用し、20 秒後に終了します。

Target \$ **./encodededcode -r ZOOM -t 20**

このデモの詳細については、encodededcode.txt ファイルおよび 3.3.1 項「Encode + Decode デモについて」を参照してください。

A.1.2 コマンドラインから実行する Encode デモ

このデモは、ペリフェラル・デバイス・ドライバからデータを取得してファイルにエンコードします。ビデオと音声のファイルがサポートされています。作成されるファイルは、ヘッダの付いていないエンコード済みデータの Raw フレームです。

Target \$ **./encode [options]**

オプション:

- ❑ **-s|--speechfile**
音声出力の送信先のファイル名を指定します。ファイル拡張子によって使用するフォーマットが識別されます。サポートされている音声アルゴリズムは、G.711 (ファイル拡張子 .g711) です。ファイルは存在していない場合は作成され、存在している場合は上書きされます。
- ❑ **-v|--videofile**
ビデオ出力の送信先のファイル名を指定します。ファイル拡張子によって使用するフォーマットが識別されます。サポートされているビデオ・アルゴリズムは、MPEG4 (ファイル拡張子 .mpeg4) および H.264 (ファイル拡張子 .264) です。ファイルは存在していない場合は作成され、存在している場合は上書きされます。
- ❑ **-b|--bitrate**
ビデオをエンコードするビット・レートを指定します。デフォルトは 4000000 bps です。
- ❑ **-t|--time**
デモを実行する秒数。デフォルトでは、時間制限はありません。

- ❑ `-l|--linein`
録音用の入力デバイスをデフォルトの「Mic In」から「Line In」に変更します。
- ❑ `-i|--interface`
これを使用すると、メイン・デモ・インターフェイスがこのデモの終了時に起動します。デフォルトでは、オフになっています。
- ❑ `-x`
S-Video 入力フォーマットを選択します。このフラグは、DVEVM 1.1 以上でのみ使用できます (DVEVM 1.0 はコンポジット・ビデオのみをサポートします)。
- ❑ `-h|--help`
このヘルプ・メッセージを出力します。

次の例は MPEG4 ビデオをエンコードし、オーディオはエンコードしません。デモは 20 秒後に終了します。

```
Target $ ./encode -v test.mpeg4 -t 20
```

次の例は 1 Mbps で H.264 ビデオをエンコードし、オーディオはエンコードしません。

```
Target $ ./encode -v test.264 -b 1048576
```

次の例は H.264 ビデオをエンコードし、G.711 音声をエンコードします。

```
Target $ ./encode -v test.264 -s test.g711
```

Encode デモを実行するには、少なくとも 1 つのビデオまたは音声ファイル、あるいはその両方を適切なファイル・フォーマットの拡張子で指定する必要があります。このデモの詳細については、`encode.txt` ファイルおよび 3.3.2 項「Encode デモについて」を参照してください。

A.1.3 コマンドラインから実行する Decode デモ

このデモはコーデック・エンジンを使用して、ファイルのデータをデコードします。これはペリフェラル・デバイス・ドライバを使用する非圧縮データを出力します。ビデオ、オーディオ、および音声のファイルがサポートされています。ファイルはデータの Raw フレームで構成されていなければなりません。

```
Target $ ./decode [options]
```

オプション:

- ❑ `-a|--audiofile`
再生するオーディオ・ファイルのファイル名を指定します。ファイル拡張子によって使用するフォーマットが識別されます。サポートされているオーディオ・アルゴリズムは、AAC (ファイル拡張子 .aac) および MPEG1 レイヤ 2 (ファイル拡張子 .mp2) です。オーディオ・ファイルと音声ファイルの両方を同時に再生することはできません。

- ❑ **-s|--speechfile**
再生する音声ファイルのファイル名を指定します。ファイル拡張子によって使用するフォーマットが識別されます。サポートされている音声アルゴリズムは、G.711（ファイル拡張子 .g711）です。オーディオ・ファイルと音声ファイルを両方とも同時に再生することはできません。
- ❑ **-v|--videofile**
再生するビデオのファイル名を指定します。ファイル拡張子によって使用するフォーマットが識別されます。サポートされているビデオ・アルゴリズムは、MPEG4（ファイル拡張子 .mpeg4）、H.264（ファイル拡張子 .264）、および MPEG2（ファイル拡張子 .m2v）です。
- ❑ **-t|--time**
デモを実行する秒数。デフォルトでは、時間制限はありません。
- ❑ **-l|--loop**
これを使用すると、ファイルが終了したときにファイルの先頭にループバックします。
- ❑ **-i|--interface**
これを使用すると、メイン・デモ・インターフェイスがこのデモの終了時に起動します。デフォルトでは、オフになっています。
- ❑ **-h|--help**
このヘルプ・メッセージを出力します。

次の例は MPEG2 ビデオをデコードし、音声もオーディオもデコードもしません。デモは 20 秒後に終了します。

```
Target $ ./decode -v data/videos/mpeg2ts_part1.m2v -t 20
```

次の例は AAC オーディオのデコードのみを行います。

```
Target $ ./decode -a data/sounds/19_44.aac
```

次の例は MPEG2 ビデオと AAC オーディオをデコードします。

```
Target $ ./decode -a data/sounds/19_44.aac
        -v data/videos/mpeg2ts_part1.m2v
```

次の例は MPEG4 ビデオと G.711 音声をデコードします。

```
Target $ ./decode -v data/sounds/dlp.mpeg4
        -s data/sounds/Input1_Alaw.g711
```

Decode デモを実行するには、少なくとも 1 つのビデオ、音声、またはオーディオ・ファイルのいずれかを指定する必要があります。ビデオと音声の両方、またはビデオとオーディオの両方を指定しても構いません。3 つを同時に指定することはできません。

このデモの詳細については、decode.txt ファイルおよび 3.3.3 項「Decode デモについて」を参照してください。

A.2 ビデオ入出力方式の変更方法

DVEVM には次の方式のビデオを入力できます。

- ☐ コンポジット [デフォルト]
- ☐ S-Video (最高画質)

さらに、3 タイプのビデオを出力できます。

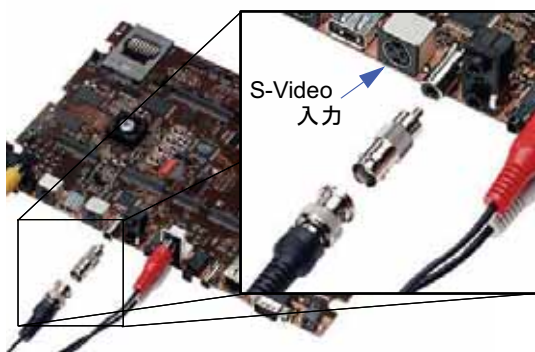
- ☐ コンポジット [デフォルト] (最低画質)
- ☐ S-Video (中間画質)
- ☐ コンポーネント (最高画質)

各種の入力と出力には大きな画質の違いがあります。しかし、DVEVM キットのケーブルはコンポジット・ビデオしかサポートしていません。S-Video またはコンポーネント・ビデオのケーブルは別途入手する必要があります。

A.2.1 S-Video 入力の使用方法

次の手順で高画質の S-Video 入力に切り替えます。

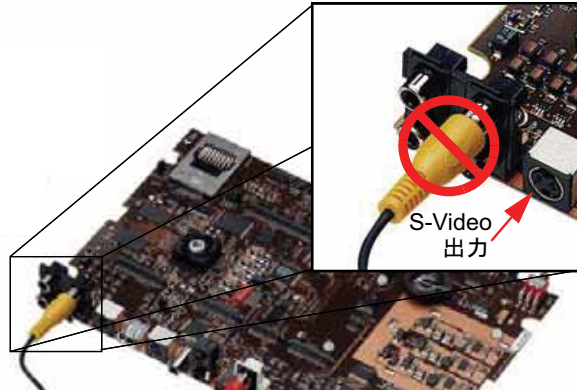
- 1) S-Video コネクタを S-Video 入力ポートに接続します。このポートは、現在使用されているコンポジット・ビデオの入力ポートのすぐ左側にあります。
- 2) Encode デモまたは Encode + Decode デモで S-Video 入力を選択するには、コマンドラインで「-x」フラグを指定します。このフラグは、DVEVM 1.1 以降でのみ使用できます (DVEVM 1.0 はコンポジット・ビデオのみをサポートします)。



A.2.2 S-Video 出力の使用方法

次の手順で高画質の S-Video 出力に切り替えます。

- 1) コンポジット・ビデオのコネクタを抜きます。次に、S-Video コネクタを S-Video 出力ポートに接続します。このポートは、現在使用されているコンポジット・ビデオの出力ポートの右側にあります。



DVEVM キットには、S-Video ケーブルは付属していません。また、S-Video 入力の可能なビデオ・ディスプレイが必要になります。

- 2) カーネルのコマンドラインでは、NTSC か PAL のいずれかと S-Video 出力フォーマットを選択して DVEVM を構成できます（4.3.6 項「PAL ビデオ・ユーザ用のブート・セットアップの構成」を参照）。たとえば、NTSC と S-Video 出力が必要な場合は、次のコマンドを使用します。

```
video=dm64xxfb:output=ntsc:format=s-video
```

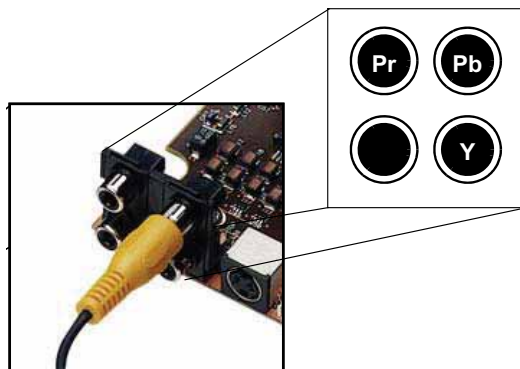
PAL と S-Video が必要な場合は、次のコマンドを使用します。

```
video=dm64xxfb:output=pal:format=s-video
```

A.2.3 コンポーネント・ビデオ出力の使用方法

次の手順で最高画質のコンポーネント・ビデオ出力に切り替えます。

- 1) コンポーネント・ビデオのコネクタを、ボードの一番左側にある枠内のコネクタに接続します。コンボジット・ビデオのように 1 つのコネクタを接続するのではなく、YPrPb コネクタを次のように接続します。



DVEVM キットには、コンポーネント (YPrPb) ビデオに使用する 3 コネクタ・ケーブルは付属していません。また、コンポーネント・ビデオ入力の可能なビデオ・ディスプレイが必要になります。

- 2) カーネルのコマンドラインでは、NTSC か PAL のいずれかとコンポーネント・ビデオ出力フォーマットを選択して DVEVM を構成できます (4.3.6 項「PAL ビデオ・ユーザ用のブート・セットアップの構成」を参照)。たとえば、NTSC とコンポーネント・ビデオ出力が必要な場合は、次のコマンドを使用します。

```
video=dm64xxfb:output=ntsc:format=component
```

PAL とコンポーネント・ビデオが必要な場合は、次のコマンドを使用します。

```
video=dm64xxfb:output=pal:format=component
```

A.3 デモ・アプリケーションを Third Party Menu に追加する方法

以降の手順を実行すると、ユーザが作成したデモを Third Party Menu に追加することができます。ユーザ・インターフェイスで一度に表示できるデモは 4 つのみです。5 つ以上のデモを追加すると、最初の 4 つがアルファベット順に表示されます。

- 1) デモ用に次のファイルを作成します。

■ logo.jpg

これはデモの説明の横に表示されるサードパーティ会社のロゴです。この画像は 50x50 のサイズの JPEG フォーマットである必要があります。

■ readme.txt

これはテキスト・ファイルです。ファイルの最初の 40 文字は、デモの簡単な説明です。デモ・インターフェイスには最大 40 文字が表示されますが、改行文字を検出すると停止します。たとえば、このファイルに「Video Phone demo」や「Network Audio demo」などと記述できます。

■ app.sh

これはユーザのデモを起動する実行可能ファイルです。このファイルはデモの実行可能ファイル自体か、またはこの実行可能ファイルを実行するシェル・スクリプトです（シェル・スクリプトの場合は、その実行可能ビットがすべてに対してセットされていることを確認してください）。スクリプトは次のようなものです。

```
#!/bin/sh
exec ./mydemoname
```

■ その他のファイル

app.sh がシェル・スクリプトの場合、ユーザのデモ実行ファイルには別の名前が付きます。また、実行ファイルで使用するデータ・ファイルやその他のファイルを用意しなければならない場合もあります。

注：デモ・アプリケーションは、相対パスを使用して実行時に必要なファイルにアクセスする必要があります。これはデモが実行される場所とは別の場所にアーカイブが展開されるためです。

- 2) 上記のリスト内のファイルをすべてアーカイブする gzip を使って圧縮された tar ファイル（末尾が .tar.gz）を作成します。たとえば、logo.jpg、readme.txt、および app.sh のファイルがある場合は、次のコマンドを使用できます。

```
tar cvzf ti_videophone.tar.gz logo.jpg readme.txt app.sh
```

この tar ファイルには、慣例として <会社名>_<デモ名>.tar.gz（ファイル名にはスペースを使用しない）と名前を付けます。たとえば、Texas Instruments が作成するテレビ電話用デモの名前は、ti_videophone.tar.gz のようになります。この名前は、すべてのデモが同一のディレクトリにインストールされるため固有でなければなりません。

必要な 3 つのファイルは、アーカイブの一番上のレベルのディレクトリ内に置く必要があります。その他のファイルはサブディレクトリに置くことができますが、デモがそのファイルへのアクセスに相対参照を使用する場合のみです。たとえば、アーカイブでは次のディレクトリ構造を使用できます。

```
|-- app.sh
|-- data
|   |-- datafile1
|   `-- datafile2
|-- logo.jpg
`-- readme.txt
```

作成したファイルのフォーマットを確認するには、Linux で次のコマンドを実行します。その結果として、「gzip compressed data」と示されます。

```
file<filename>.tar.gz
```

- 3) ユーザのアーカイブをターゲットのインストール・ディレクトリの「thirdpartydemos」サブディレクトリに置きます。これは DVEVM ソフトウェアがターゲット・ファイル・システムにインストールされた場所です。ターゲットのインストール・ディレクトリのデフォルトは /opt/dvevm です。したがって、デモ・アーカイブのデフォルトの場所は /opt/dvevm/thirdpartydemos です。この場所にはアーカイブの内容を展開しないでください。デモが実行されるたびに舞台裏で展開が実行されます。

A.4 TFTP サーバのセットアップ方法

次のコマンドを使用して、TFTP サーバがセットアップされているかどうかを確認することができます。

```
host $ rpm -q tftp-server
```

セットアップされていない場合の手順は、次のとおりです。

- 1) MontaVista Linux のデモ版（4.3.1 項を参照）をインストールしていない場合は、インターネット上の多くの場所からご使用の Linux ホスト用の TFTP サーバをダウンロードできます。「tftp-server」を検索してください。
- 2) TFTP をインストールするには、次のコマンドを使用します。ここで、`-#.##-#` はファイル名のバージョン番号の部分です。

```
host $ rpm -ivh /db/ztree/useracct/tftp-server-#.##-#.rpm
```

次の出力が表示されます。

```
warning: /db/ztree/useracct/tftp-server-#.##-#.rpm:
V3 DSA signature: NOKEY, key ID 4f2a6fd2
Preparing... ##### [100%]
1:tftp-server ##### [100%]
```

- 3) 次のコマンドを使用して、TFTP がインストールされていることを確認します。

```
host $ /sbin/chkconfig --list
```

次の出力が表示されます。TFTP がリストの最後にあることに注意してください。

```
xinetd based services:
  krb5-telnet:      off
  rsync:            off
  eklogin:          off
  gssftp:           off
  klogin:           off
  ...
  sgi_fam:          on
  rexec:            off
  rlogin:           off
  rsh:              on
  telnet:           on
  tftp:             off
```

TFTP サーバを起動しないようにするには、次のコマンドを使用します。

```
/sbin/chkconfig tftp on
```

TFTP ファイルの処理に使用されるデフォルトのルート場所は `/tftpboot` です。

A.5 その他のブート方法

DVEVM のデフォルト構成では、ボードのハードディスク・ドライブのファイル・システムを使用してフラッシュからブートします。ボードをブートする他の方法は、次のとおりです。

- ❑ ハードディスク・ドライブのファイル・システムによる TFTP ブート (A.5.2 項)
- ❑ NFS ファイル・システムによるフラッシュ・ブート (A.5.3 項)
- ❑ NFS ファイル・システムによる TFTP ブート (A.5.4 項)

4.3.6 項では、PAL ビデオ・モードと NTSC ビデオ・モードのブート方法について説明しています。

以降の節では、各ブート方法を有効にするための環境変数の設定方法を示します。

これらのモードのいずれかでブートする手順は、次のとおりです。

- 1) DVEVM ボードの電源を入れ、コンソール・ウィンドウでキーを押して自動ブートシーケンスを終了します (2.2 節)。
- 2) 以降の節に記述されている、使用したいブート・モードの環境変数をセットします。
- 3) これらの設定をデフォルトとして今後も使用したい場合は、その環境を保存します。

```
EVM # saveenv
```

- 4) ユーザが行った設定を使用してボードをブートします。

```
EVM # boot
```

A.5.1 ボードのハードディスク・ドライブのファイル・システムを使用したフラッシュからのブート

これがすぐに使用できるデフォルトのブート構成です。

このモードでブートするには、自動ブートシーケンス中断後に次のパラメータをセットします。

```
EVM # setenv bootcmd bootm 0x2050000
EVM # setenv bootargs mem=120M console=ttyS0,115200n8
      root=/dev/hda1 rw noinitrd ip=dhcp
```

たとえば、このモード用にセットされる環境変数は次のようになります。

```
bootargs=mem=120M console=ttyS0,115200n8 root=/dev/hda1 rw
noinitrd ip=dhcp
bootdelay=3
baudrate=115200
bootfile=library/davinci/0.4.2/uImage (NOT USED)
serverip=192.168.160.71 (NOT USED)
bootcmd=bootm 0x2050000
stdin=serial
stdout=serial
stderr=serial
ethaddr=00:0e:99:02:51:46
```

ブート時、次の行でブート・モードを確認します。

```
## Booting image at 02050000 ...
```

A.5.2 ボードのハードディスク・ドライブのファイル・システムを使用した TFTP 経由のブート

このモードでブートするには、自動ブートシーケンス中断後に次のパラメータをセットします。

```
EVM # setenv bootcmd 'dhcp;bootm'
EVM # setenv bootargs mem=120M console=ttyS0,115200n8
      root=/dev/hda1 rw noinitrd ip=dhcp
EVM # setenv serverip <your_tftp_server_ip_address>
EVM # setenv bootfile <path_on_tftpserver>/uImage
```

たとえば、このモード用にセットされる環境変数は次のようになります。

```
bootargs=mem=120M console=ttyS0,115200n8 root=/dev/hda1 rw
noinitrd ip=dhcp
bootdelay=3
baudrate=115200
stdin=serial
stdout=serial
stderr=serial
ethaddr=00:0e:99:02:51:46
serverip=xxx.xxx.xxx.xxx (your tftp server)
bootcmd=dhcp;bootm
bootfile=tftp_path/uImage
```

ブート時、次の行でブート・モードを確認します。

```
TFTP from server 192.168.160.71; our IP address is
192.168.161.186
Filename 'library/davinci/0.4.2/uImage'.
...
## Booting image at 80700000 ...
```

A.5.3 NFS ファイル・システムを使用したフラッシュからのブート

このモードでブートするには、自動ブートシーケンス中断後に次のパラメータをセットします。

```
EVM # setenv bootcmd 0x2050000
EVM # setenv nfshost 192.168.160.57
EVM # setenv rootpath /exports/useracct
EVM # setenv bootargs console=ttyS0,115200n8 noinitrd rw
    ip=dhcp
    root=/dev/nfs nfsroot=$(nfshost):$(rootpath),nolock
    mem=120M
```

たとえば、このモードにセットされる環境変数は次のようになります。

```
bootdelay=3
baudrate=115200
bootfile=library/davinci/0.4.2/uImage
serverip=192.168.160.71
stdin=serial
stdout=serial
stderr=serial
ethaddr=00:0e:99:02:51:46
bootcmd=0x2050000
nfshost=192.168.160.57
rootpath=/exports/useracct
bootargs=console=ttyS0,115200n8 noinitrd rw ip=dhcp
root=/dev/nfs nfsroot=192.168.160.57:/ex
ports/useracct,nolock mem=120M eth=00:0e:99:02:51:46
```

ブート時、次の行でブート・モードを確認します。

```
## Booting image at 02050000 ...
...
Starting kernel ...
...
VFS: Mounted root (nfs filesystem).
```

A.5.4 NFS ファイル・システムを使用して TFTP 経由でブートする方法

このモードでブートするには、自動ブートシーケンス中断後に次のパラメータをセットします。

```
EVM # setenv bootcmd 'dhcp;bootm'
EVM # setenv serverip 192.168.160.71
EVM # setenv bootfile library/davinci/0.4.2/uImage
EVM # setenv rootpath /exports/useracct
EVM # setenv nfshost 192.168.160.57
EVM # setenv bootargs console=ttyS0,115200n8 noinitrd rw
      ip=dhcp
      root=/dev/nfs nfsroot=$(nfshost):$(rootpath),nolock
      mem=120M
```

たとえば、このモードにセットされる環境変数は次のようになります。

```
bootdelay=3
baudrate=115200
stdin=serial
stdout=serial
stderr=serial
ethaddr=00:0e:99:02:51:46
bootcmd=dhcp;bootm
serverip=192.168.160.71
bootfile=library/davinci/0.4.2/uImage
rootpath=/exports/useracct
nfshost=192.168.160.57
bootargs=console=ttyS0,115200n8 noinitrd rw ip=dhcp
root=/dev/nfs nfsroot=192.168.160.57:/ex
ports/useracct,nolock mem=120M eth=00:0e:99:02:51:46
```

ブート時、次の行でブート・モードを確認します。

```
TFTP from server 192.168.160.71; our IP address is
192.168.161.186
Filename 'library/davinci/0.4.2/uImage'.
...
Starting kernel ...
...
VFS: Mounted root (nfs filesystem).
```

A.6 Windows でのコンポーネントのインストール方法

Windows バージョンの MontaVista Linux ツールをすでに所有している場合には、次の Microsoft Windows インストーラが DVSDK で提供されています。

❑ DVEVM: mvl_lsp_setupwin32_#_##_###.exe

❑ DVEVM: dvevm_setupwin32_#_##_###.exe

DVEVM には Windows バージョンの MontaVista Linux ツールは組み込まれていません。DaVinci 開発に Windows ホストを使用したい場合は、Texas Instruments から DVSDK を入手する必要があります。また、Windows バージョンの MVL Target (ターゲット側のコードを生成するための Windows で稼働するクロス・コンパイラ) も必要です。

DaVinci 開発に Windows ホストをセットアップする手順は、次のとおりです。

- 1) Windows インストーラをすべて上記の順序で実行します。DVEVM ソフトウェアは、MontaVista Linux ツールがすでにインストール済みであることを想定しています。
- 2) インストーラを実行する際 (MVL バッシュ・シェルを使用している場合でも)、Windows 形式のバックスラッシュを使用した絶対パスを指定します。たとえば、C:\mvcyg4.0\home\useracct のように指定します。
- 3) テキスト・エディタを使用して、Rules.make ファイルを編集します。次のように変更します。

■ MVTOOL_PREFIX: GNU コンパイラ・ツールの前にプレフィックスを追加するように指定します。先頭に C: を付けて UNIX フォーマット (スラッシュ) を使用します。たとえば次のとおりです。

```
C:/mvcyg4.0/opt/mv_pro_4.0/montavista/pro/devkit/arm/v5t_le/bin/arm_v5t_le-
```

■ EXEC_DIR: UNIX フォーマット (スラッシュ) を使用して、生成される実行可能ファイルとデータをコピーする場所を指定します。たとえば、C:/mvcg4.0/opt/dvevm のように指定します。

A.7 DSP/BIOS Link の再ビルド方法

以下の手順で DSP/BIOS Link パッケージを再ビルドします(バッシュ・シェルを使用していることを想定しています)。

- 1) `BASE_BUILD` 変数と `BASE_CGTOOLS` 変数が正しい適切な場所を示すように、`davinci_mvlpro4.0.mk` ファイルを編集します。このファイルは、
~/dvevm_#_###/dsplink_1_###_###/packages/dsplink/make/Linux/ ディレクトリにあります。
- 2) `DSPLINK` 環境変数を「dsplink」ディレクトリへの絶対パスとなるように定義します (バッシュ・シェルでは `export`、`tcsh` シェルでは `setenv` を使用します)。たとえば次のとおりです。

```
host $ export DSPLINK=/home/useracct/dvevm_#_###/dsplink_1_30_08_02/packages/dsplink
```

- 3) Linux ビルド・スクリプトのディレクトリに移動します。

```
host $ cd $DSPLINK/etc/host/scripts/Linux
```

- 4) DSP/BIOS Link を次のようにビルドします。

```
host $ sh -f buildmodule.sh
```

- 5) 再ビルドされたカーネル・モジュールは `dsplinkk.ko` と呼ばれます。これは `$DSPLINK/gpp/export/BIN/Linux/Davinci/RELEASE/` ディレクトリにあります。

A.8 DVEVM ハードディスク・ドライブの復元および更新方法

ここでは、MontaVista ファイル・システムやデモなどを含む DVEVM ハードディスク・ドライブ (HDD) のすべてのファイルを復元および更新する方法について説明します。この復元手順を使用すると、ボードの HDD のデータに何か起こったときにそのボードを元の状態に戻すことができます。

ここでは、DVEVM で NFS root マウントを実行するのに必要なソフトウェアを使用して、4.3.4 項および 4.3.5 項の説明どおりにホスト Linux ワークステーションを構成済みであることを想定しています。

ここでは、U-Boot は常にターゲットのフラッシュ・メモリの先頭位置 (アドレス 0x02000000) にあります。同様に、uImage (Linux カーネル・プログラム) は、ターゲットのフラッシュ・メモリのアドレス 0x02050000 からブートします。

A.8.1 システムのセットアップ

ハードディスク・ドライブの復元または更新を行うには、事前に次のシステムのセットアップ手順を実行している必要があります。

- 1) 「CS2 SELECT」というラベルが付いているジャンパ J4 を調べます。FLASH が選択されていることを確認します。
- 2) ホスト・ワークステーションのイーサネット・ポートをルータに接続します。ルータ内で稼働している DHCP サーバ経由で動的に IP アドレスを取得するようにホストのイーサネット・ポートを構成します。
- 3) ターゲット DVEVM のイーサネット・ポートを同じルータの別のポートに接続します。これでホスト・ワークステーションとのネットワーク接続が確立されます。
- 4) RS-232 ケーブルをターゲット DVEVM ボードの UART0 ポートからホスト・ワークステーションに接続します。
- 5) ホスト・ワークステーション上で、次の特性を使用してターゲット DVEVM ボードに対するターミナル・セッションを開きます。
 - bps: 115200
 - データ・ビット: 8
 - パリティ: なし
 - ストップ・ビット: 1
 - フロー制御: なし

たとえば、ターミナル・セッションを作成する場合、MS Windows では HyperTerminal または TeraTerm、Linux では Minicom または C-Kermit を使用できます。

- 6) ホスト・ワークステーション上で NFS サーバを起動します。本書では、ターゲット DVEVM が root マウントに使用できるファイル・システムが、ホストのパス /home/user/workdir/filesys に入っていることを想定しています。

A.8.2 DVEVM の NFS root マウントの構成

DVEVM の NFS root マウントを構成する手順は、次のとおりです。

- 1) Boot Switches (S3) を 1011111110 に構成します。これは DVEVM の中央にあるスイッチの赤いバンクです。
- 2) DVEVM に電源を入れて、いずれかのキーを押して U-Boot に入ります。
- 3) 次のように bootcmd を構成して、フラッシュ経由で Linux カーネルをブートするようにします。

```
EVM # setenv bootcmd bootm 0x2050000
```

- 4) 次のように bootargs を構成して、NFS からファイル・システムを root 権限でマウントするようにします。

```
EVM # setenv bootargs 'console=ttyS0,115200n8 noinitrd rw ip=dhcp
root=/dev/nfs nfsroot=192.168.1.102:/home/user/workdir/filesys,nolock mem=120M'
```

このコマンドの nfsroot オプションは、ホスト・ワークステーションの IP アドレスを使用します。必ずこの IP アドレス 192.168.1.102 を、ご使用のホスト Linux ワークステーションの実際のアドレスに置き換えてください。

- 5) オプション: U-Boot パラメータを出力します。

```
EVM # printenv
```

- 6) U-Boot パラメータを保存します。

```
EVM # saveenv
```

- 7) ホスト Linux ワークステーションの NFS から DVEVM をブートします。

```
EVM # boot
```

- 8) 「root」として MontaVista Linux にログインします。

A.8.3 DVEVM ハードディスク・ドライブの復元

DVEVM ハードディスク・ドライブ (HDD) は、ターゲット DVEVM HDD のパーティションまたはホスト Linux ワークステーションのファイル・システムから復元できます。これらの 2 通りの方法を両方とも実行する必要はありません。どちらも同じ結果になります。

DVEVM HDD の復元には 10 ～ 15 分を要します。復元スクリプトは、600 MB の圧縮データを圧縮解除し、そのデータを /dev/hda1 パーティションにロードします。

ハードディスク・ドライブの復元プロセスが完了したら、必ず DVEVM を再起動して、ローカル HDD 経由で root 権限でマウントするように U-Boot を構成してください。このタイプのブートの手順は、A.5.1 項「ボードのハードディスク・ドライブのファイル・システムを使用したフラッシュからのブート」に記述されています。

A.8.3.1 ターゲット DVEVM HDD パーティションからの復元方法

HDD 自体の復元パーティションから HDD を復元する手順は、次のとおりです。

- 1) HDD 復元パーティションをマウントするためのディレクトリを作成します。

```
EVM # mkdir /mnt/restore
```

- 2) HDD 復元パーティションをマウントします。

```
EVM # mount -t ext3 /dev/hda2 /mnt/restore
```

- 3) Linux の date 変数を今日の日付にセットします。日付が大幅にずれている場合は、ターゲット・ファイル・システムのインストールで一連の警告が生成されます。

```
EVM # date MMDDHHMMCCYY
```

たとえば、2006 年 4 月 18 日 9:00 am の場合は 041809002006 と入力します。

- 4) ディレクトリを /mnt/restore に変更します。

```
EVM # cd /mnt/restore
```

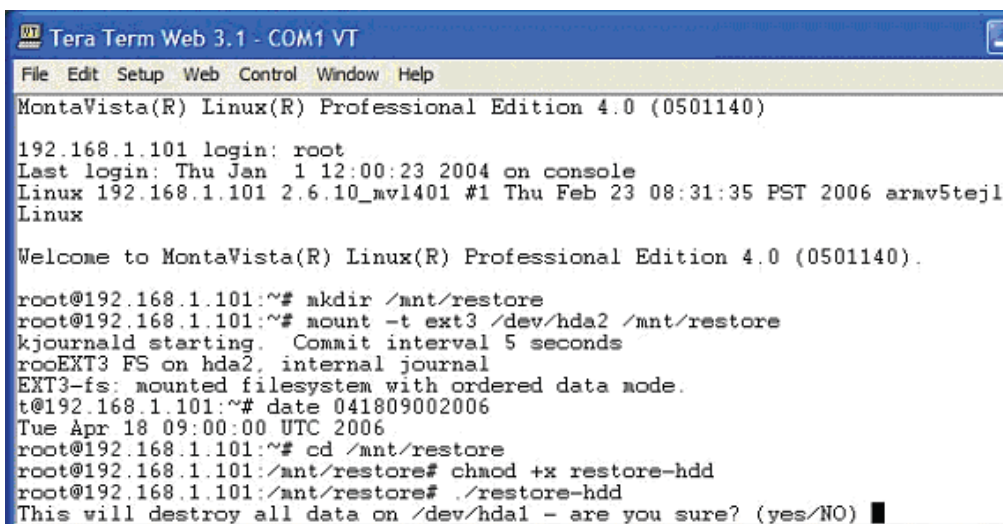
- 5) スクリプトの実行権限を付加します。

```
EVM # chmod +x restore-hdd
```

- 6) 復元スクリプトを実行します。

```
EVM # ./restore-hdd
```

- 7) スクリプトが確認を求めます。"This will destroy all data on /dev/hda1 - are you sure?" (これにより /dev/hda1 のすべてのデータが破棄されます。よろしいですか?) **yes** と入力します。



```

Tera Term Web 3.1 - COM1 VT
File Edit Setup Web Control Window Help
MontaVista(R) Linux(R) Professional Edition 4.0 (0501140)
192.168.1.101 login: root
Last login: Thu Jan 1 12:00:23 2004 on console
Linux 192.168.1.101 2.6.10_mvl401 #1 Thu Feb 23 08:31:35 PST 2006 armv5tej1
Linux
Welcome to MontaVista(R) Linux(R) Professional Edition 4.0 (0501140).
root@192.168.1.101:~# mkdir /mnt/restore
root@192.168.1.101:~# mount -t ext3 /dev/hda2 /mnt/restore
kjournald starting. Commit interval 5 seconds
rooEXT3 FS on hda2, internal journal
EXT3-fs: mounted filesystem with ordered data mode.
t@192.168.1.101:~# date 041809002006
Tue Apr 18 09:00:00 UTC 2006
root@192.168.1.101:~# cd /mnt/restore
root@192.168.1.101:/mnt/restore# chmod +x restore-hdd
root@192.168.1.101:/mnt/restore# ./restore-hdd
This will destroy all data on /dev/hda1 - are you sure? (yes/NO) █

```

- 8) HDD の復元が完了したら、DVEVM をシャットダウンします。
EVM # **halt**
- 9) ターミナル・ウィンドウに「Power down」というメッセージが出力されたら、DVEVM の電源を安全に切ることができます。
- 10) DVEVM を再起動して、ローカル HDD 経由で root 権限でマウントするように U-Boot を構成します。A.5.1 項「ボードのハードディスク・ドライブのファイル・システムを使用したフラッシュからのブート」に記述されている手順を実行します。

A.8.3.2 ホスト Linux ワークステーションのファイル・システムからの復元

ここでは、DVEVM ソフトウェアがホスト Linux の /workdir/filesys/restore ディレクトリにインストール済みであることを想定しています。NFS のマウント後、これはターゲット DVEVM の /restore に相当します。

ホスト Linux ワークステーションの復元ディレクトリから HDD を復元する手順は、次のとおりです。

- 1) root として DVEVM にログインします。
- 2) /restore ディレクトリに移動します。
EVM # **cd /restore**
- 3) Linux の date 変数を今日の日付にセットします。日付が大幅にずれている場合は、ターゲット・ファイル・システムのインストールで一連の警告が生成されます。
EVM # **date MMDDHHMMCCYY**
たとえば、2006 年 4 月 18 日 9:00 am の場合は 041809002006 と入力します。
- 4) restore-hdd スクリプトの実行権限を付加します。
EVM # **chmod +x restore-hdd**
- 5) /restore ディレクトリにあるこのスクリプトを実行します。
EVM # **./restore-hdd**
- 6) スクリプトが確認を求めます。"This will destroy all data on /dev/hda1 - are you sure?" (これにより /dev/hda1 のすべてのデータが破棄されます。よろしいですか?) **yes** と入力します。
- 7) HDD の復元が完了したら、DVEVM をシャットダウンします。
EVM # **halt**
- 8) ターミナル・ウィンドウに「Power down」というメッセージが出力されたら、DVEVM の電源を安全に切ることができます。
- 9) DVEVM を再起動して、ローカル HDD 経由で root マウントするように U-Boot を構成します。A.5.1 項「ボードのハードディスク・ドライブのファイル・システムを使用したフラッシュからのブート」に記述されている手順を実行します。

NFSファイルシステムを介してCDIに付属するoverlay.tar.gzをマニュアルで保存し、インストールした場合、デモは正常に動作しません。このような方法はドキュメントに記載されていませんし、お勧めいたしません。問題は下記の方法でsビットをオフにすることで解決できます。

```
EVM # cd /opt/dvevm
EVM # chmod u-s ${install}/encode ${install}/decode ${install}/encodedecode \
      ${install}/interface
```

A.8.4 更新 Web サイトからの最新 DVEVM ハードディスク・ドライブ・ソフトウェアのダウンロード

DVEVM 更新 Web サイトから、最新の DVEVM HDD ソフトウェアをダウンロードしてインストールできます。この場合の手順は次のとおりです。

- 1) 次の場所から最新の DVEVM ソフトウェア更新データ (dvevm_setuplinux_#_##_##_#.bin) をホストのファイル・システムにダウンロードします。

<https://www-a.ti.com/extranet/cm/product/dvevmmsw/dspswext/general/homepage.shtml>

- 2) ホストのファイル・システムからターゲットの DVEVM HDD を復元するには、A.8.3.2 項「ホスト Linux ワークステーションのファイル・システムからの復元」に記述されている手順を実行してください。

索引

A

AAC オーディオ 3-8, A-2, A-4
ARM 側の開発 4-2
ARM9 1-3

B

bin ファイル 4-6

C

C64+ DSP 1-3
CD
 ファイル内容 4-2
 マウント 4-5
CD-ROM 1-3
Code Search ボタン 3-2
COM ポート 2-6
CPU の負荷 3-4

D

DaVinci テクノロジー 1-2
 コミュニティ 1-5
Decode デモ 3-3, 3-8
 コマンドライン A-4
DHCP サーバ 3-2
DISPLAY 環境変数 4-5
DSP 4-4
 ブラック・ボックス 4-2
DSP/BIOS Link 4-4
DVD ボタン 3-2
DVEVM 1-2
 インストール、ソフトウェア 4-7
DVEVM ソフトウェア
 再ビルド 4-12
dvevm_setupwin32_#.exe インストーラ A-16
DVSDK 4-2, 4-3

E

Encode + Decode デモ 3-3, 3-5
 コマンドライン A-2
Encode デモ 3-3, 3-6
 コマンドライン A-3
ESD 防止 2-3
EVM # プロンプト 2-6, 4-3
EXEC_DIR 環境変数
 環境変数
 EXEC_DIR A-16
exports ファイル 4-9

G

G.711 音声 3-6, 3-8, A-2, A-3, A-5

H

H.264 ビデオ 3-5, 3-6, 3-8, A-2, A-3, A-5
HDD A-18
 更新 A-22
 復元 A-18, A-20
host \$ プロンプト 4-3
HyperTerminal 2-6

I

Info/Select ボタン 3-4
IP アドレス、動的な 3-2

L

LCD ディスプレイ 1-3, 2-2
Linux 4-4
 インストール 4-6
 カーネル 4-12
 バージョン、サポートされる 4-6
Linux Support Package 4-12

M

Microsoft Windows 4-8, A-16
modules.tar.gz ファイル 4-2
MontaVista Linux 4-2
 完全版 4-3
 デモ版 4-3
MontaVista Linux ツール Windows 版 A-16
MPEG1 レイヤ 2 オーディオ 3-8, A-2, A-4
MPEG2 ビデオ 3-8, A-2, A-5
MPEG4 ビデオ 3-6, 3-8, A-2, A-3, A-5
mvl_lsp_setupwin32_#.exe インストーラ A-16
MVTOOL_PREFIX 環境変数
 環境変数
 MVTOOL_PREFIX A-16

N

NFS サーバ 4-8
 テスト 4-9
 ブート構成 A-14, A-15
NTSC ビデオ 2-2, 4-10

O

OSD 切り換え 3-4
OSD の透過性 3-4
OSD の表示と非表示 3-4
overlay.tar.gz ファイル 4-2

P

PAL ビデオ 2-2, 4-10
PATH 環境変数 4-11, 4-12
Pause ボタン 3-4
Play ボタン 3-4
Power ボタン 3-3

R

Record ボタン 3-4
Red Hat Enterprise Linux 4-6
restore ディレクトリ 4-2
Rules.make ファイル 4-12

S

Spectrum Digital 社の Web サイト 1-4
Stop ボタン 3-4
SuSe Workstation 4-6
S-Video A-3, A-4, A-6
 出力 A-7
 入力 A-6

T

target \$ プロンプト 4-3
TFTP
 サーバ A-11
 転送、ファイルのボードへの 4-14
 ブート構成 A-13, A-15
Third Party Menu 3-3, A-9
TMS320DM6446 1-3

U

U-Boot ユーティリティ 4-12
u-boot.bin ファイル 4-2
uImage ファイル 4-2
uImage ブート・ファイル 4-14

V

videostd 環境変数 4-10
VISA API 4-4

W

Windows 4-8, A-16
 MontaVista Linux ツール A-16

X

xDAIS 4-4
xDM 4-4

Y

YPrPb A-8

あ

アプリケーション 4-4
アプリケーションの実行方法 3-3

い

イーサネット 2-4
 セットアップ 2-5
インストール
 DVEVM ソフトウェア 4-7
 Linux ソフトウェア 4-6
 Windows ホスト A-16
 ハードウェア 2-2

お

オーディオ・ケーブル 2-3

か

カメラ 1-3, 2-3

き

キットの内容 1-3

く

クロス・コンパイラ A-16
クロック・バッテリー 1-4

け

ケーブル 1-3
 接続方法 2-2

こ

更新 A-22
コーデック・エンジン 3-4, 3-6, 4-4
コマンドライン・デモ A-2
コマンド・プロンプト 4-3
コンソール・ウィンドウ 2-6
コンポーネント・ビデオ A-6, A-8
コンボジット・ビデオ A-6

さ

再ビルド
 DVEVM ソフトウェア 4-13
 Linux カーネル 4-12

し

シリアルケーブル 2-6
シリアル接続 2-5

す

スタンドアロン・デモ 3-2

せ

静電気防止 2-2
赤外線リモート 1-3, 3-2
 コードのリセット方法 3-2

そ

ソフトウェア 4-2
 インストール 4-6
 コンポーネント 1-3, 4-4

て

ディスプレイ 1-3, 2-2
データ・ファイル 4-8
デジタルカメラ 1-3
テスト・プログラム 4-11
デモ 3-2
 コマンドライン A-2
デモの終了 3-3
電源 2-5
電源ケーブル 2-4

と

動的な IP アドレス 3-2

は

ハードディスク・ドライブ 1-3, A-18

ブート構成 A-12, A-13
復元 4-2
バッテリー 1-4, 3-2

ひ

ビデオカメラ 1-3, 2-3
電源 2-3
ビデオ・ケーブル 2-2
ビデオ・フォーマット A-6
ビルド環境 4-11

ふ

ファイル
CD 上 4-2
Decode デモ 3-8
Encode デモ 3-6
ファイル拡張子 3-6, 3-8, A-2, A-3, A-5
ファイル・システム 4-8
ブート構成 A-12
NFS 4-10
TFTP、NFS A-15
TFTP、ハードディスク・ドライブ A-13
標準 3-2
フラッシュ、NFS A-14
フラッシュ、ハードディスク・ドライブ A-12
ブートシーケンス A-12
復元、HDD A-18
フラッシュ・メモリ
ブート構成 A-12, A-14
ブロック図 1-4
プロンプト 4-3

へ

ペリフェラル 1-4, 2-2

ほ

ポート 2-4
ホスト・プラットフォーム
Windows A-16

ま

マイクrofオン 1-3, 2-4

マルチメディア・ペリフェラル 1-4

も

モニタ 2-2

や

矢印ボタン 3-3

り

リモート・コントロール 1-3, 3-2
コードのリセット方法 3-2
リンク、DSP/BIOS 4-4

れ

例 3-2

日本テキサス・インスツルメンツ株式会社

本 社 〒160-8366 東京都新宿区西新宿 6 丁目 24 番 1 号 西新宿三井ビルディング 3 階 ☎03(4331)2000(番号案内)

西日本ビジネスセンター 〒530-6026 大阪市北区天満橋 1 丁目 8 番 30 号 OAP オフィスタワー 26 階 ☎06(6356)4500(代 表)

名古屋オフィス 〒460-0003 名古屋市中区錦 2 丁目 4 番 3 号 錦パークビル 7 階 ☎052(232)5601(代 表)

工場 大分県・日出町／茨城県・美浦村

研究開発センター 茨城県・つくば市（筑波テクノロジー・センター）／神奈川県・厚木市（厚木テクノロジー・センター）

■お問い合わせ先

日本 TI プロダクト・インフォメーション・センター（PIC）————— FAX ☎0120-81-0036

URL: <http://www.tij.co.jp/pic/>

DVEVM

入門マニュアル

第 1 版 2007 年 1 月

第 2 版 2007 年 4 月

発行所 **日本テキサス・インスツルメンツ株式会社**

〒160-8366

東京都新宿区西新宿 6-24-1（西新宿三井ビルディング）

